L Number	Hits	Search Text	DB	Time stamp
63	11	(short$4 minimum) near5 delay with (up$1link down$1link)	USPAT; EPO	2003/11/14 15:18
64	1	(short$4 minimum) near5 delay with (up$1link down$1link)	JPO; IBM_TDB	2003/11/14 15:18
65	14	time near2 wraparound	USPAT; EPO	2003/11/14 15:20
66	19	(time near2 wrap$1around) not counter	USPAT; EPO	2003/11/14 15:20
67	38	time near2 wrap$1around	USPAT; EPO	2003/11/14 16:05
68	6	clock near2 wrap$1around	USPAT; EPO	2003/11/14 15:25
69	91	wrap$1around with error$1	USPAT; EPO	2003/11/14 15:26
70	18	wrap$1around with error$1 same (time timestamp)	USPAT; EPO	2003/11/14 15:28
71	2	wrap$1around with error$1 same (time timestamp)	JPO; IBM_TDB	2003/11/14 15:28
79	385	(length duration) with wrap$1around	USPAT; EPO	2003/11/14 15:30
80	190	(length duration) near6 wrap$1around	USPAT; EPO	2003/11/14 15:30
81	28	(length duration) near6 wrap$1around same (clock time)	USPAT; EPO	2003/11/14 15:31
82	5	(length duration) near5 (measure$4 calculat$4) with wrap$1around	USPAT; EPO	2003/11/14 15:32
83	54	(length duration) near5 (measure$4 calculat$4) with (wrap$3 adj around)	USPAT; EPO	2003/11/14 15:33
84	2	(length duration) near5 (measure$4 calculat$4) with (wrap$3 adj around) and g06f$.ipc.	USPAT; EPO	2003/11/14 15:35
85	3	(length duration time) near5 (measure$4 calculat$4) with (wrap$3 adj around) and g06f$.ipc.	USPAT; EPO	2003/11/14 15:35
86	32	difference with wraparound	USPAT; EPO	2003/11/14 15:41
87	62	difference with wrap$1around	USPAT; EPO	2003/11/14 15:41
88	5	difference with wrap$1around with time	USPAT; EPO	2003/11/14 15:56
89	22	wrap$1around near time	USPAT; EPO	2003/11/14 15:56
90	17	time adj wrap$1around	USPAT; EPO	2003/11/14 15:58
91	130	(time near5 resets) with (length duration)	USPAT; EPO	2003/11/14 16:00
92	7	(time near5 difference) with (wrap$1around (wrap$4 adj around))	USPAT; EPO	2003/11/14 16:01
93	1756	time with ((wrap$4 adj around) wrap$1around)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:27
94	1677	time with ((wrap$4 adj around) wrap$1around)	USPAT; EPO	2003/11/14 16:12
95	11	time$1stamp$1 with ((wrap$4 adj around) wrap$1around)	USPAT; EPO	2003/11/14 16:22
96	0	NTP with ((wrap$4 adj around) wrap$1around)	USPAT; EPO	2003/11/14 16:24
97	6	(RTD (round adj time adj delay)) with (quickest fastest shortest)	USPAT; EPO	2003/11/14 16:24
98	0	(RTD (round adj time adj delay)) with (quickest fastest shortest)	JPO; IBM_TDB	2003/11/14 16:24
99	1	NTP with ((wrap$4 adj around) wrap$1around)	JPO; IBM_TDB	2003/11/14 16:24
100	65	(time timestamp$1) with ((wrap$4 adj around) wrap$1around) with speed	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:29
101	1	(time timestamp$1) with ((wrap$4 adj around) wrap$1around) with (RTD (round adj trip adj delay))	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:45

102	976	(minimum shortest) near5 delay with (calculat$4 determin$4)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:45
103	8	(minimum shortest) near5 delay with (calculat$4 determin$4) same (RTD "round trip delay")	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:48
104	0	(minimum shortest) near5 delay with (calculat$4 determin$4) same (up$1link down$1link)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:48
105	745	(minimum shortest) near5 delay with (determin$4)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:48
106	4	(minimum shortest) near5 delay with (determin$4) same (up$1link down$1link RTD "round trip delay")	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:51
107	114	(minimum shortest) near5 delay with (determin$4) same (transmi$6)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:51
108	55	(minimum shortest) near5 delay with (determin$4) with (transmi$6)	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:53
109	325	(minimum shortest) with (up$1link down$1link RTD "round trip delay")	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:54
110	31	(determin$5 calculat$4 estimat$4 measur$4) near5 (minimum shortest) with (up$1link down$1link RTD "round trip delay")	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:57
111	31	(determin$5 calculat$4 estimat$4 measur$4) near5 (minimum shortest fastest) with (up$1link down$1link RTD "round trip delay")	USPAT; EPO; JPO; IBM_TDB	2003/11/14 16:57
152	21	"round trip delay"	JPO	2003/11/14 17:26
-	11	up$1load$4 with down$1load$4 with delay	USPAT; EPO	2003/11/14 15:09
-	0	(determin$4 near5 delay$1) with (up$1load$3 down$1load$3 up$1link down$1link) same handshake	USPAT; EPO	2003/11/10 13:12
-	37	(determin$4 near5 delay$1) with (up$1load$3 down$1load$3 up$1link down$1link)	USPAT; EPO	2003/11/10 13:12
-	37	(determin$4 near5 (latency delay$1)) with (up$1load$3 down$1load$3 up$1link down$1link)	USPAT; EPO	2003/11/10 13:53
-	0	6366762.URPN.	USPAT	2003/11/10 13:28
-	7	("5280629" | "5515062" | "5943606" | "6047161" | "6107959" | "6137441" | "6298238").PN.	USPAT	2003/11/10 13:28
-	5	(latency delay$1) with (up$1load$3 down$1load$3 up$1link down$1link) same timestamp$1	USPAT; EPO	2003/11/10 14:03
-	7	RTD with timestamp$1	USPAT; EPO	2003/11/10 14:03
-	9	RTD same timestamp$1	USPAT; EPO	2003/11/10 14:04
-	28	(RTD (round adj trip adj delay)) same timestamp$1	USPAT; EPO	2003/11/10 14:28
-	0	6545979.URPN.	USPAT	2003/11/10 14:16
-	9	("5260935" | "5339311" | "5471631" | "5483523" | "5521907" | "5793976" | "5812528" | "6023455" | "6097699").PN.	USPAT	2003/11/10 14:16
-	1361	(RTD (round adj trip adj delay)) same (fourth (time timestamp$1))	USPAT; EPO	2003/11/10 14:28
-	7	(RTD (round adj trip adj delay)) same (fourth near5 (time timestamp$1))	USPAT; EPO	2003/11/10 15:00
-	27	(up$1load$3 down$1load$3 up$1link down$1link) same (fourth near5 (time timestamp$1))	USPAT; EPO	2003/11/10 15:08

-	17	(up$1load$3 down$1load$3 up$1link down$1link) same (fourth near3 (time timestamp$1))	USPAT; EPO	2003/11/10 15:02
-	264	((up$1load$3 down$1load$3 up$1link down$1link) near5 delay) same (time timestamp$1)	USPAT; EPO	2003/11/10 16:28
-	67	((up$1load$3 down$1load$3 up$1link down$1link) near5 delay) same (time timestamp$1) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:42
-	1508	((transmi$6) near5 delay) same (time timestamp$1) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:39
-	32	((transmi$6) near5 delay) same (time timestamp$1) same subtract$4 and g06f$.ipc.	USPAT; EPO	2003/11/10 15:39
-	17	((transmi$6) near5 delay) same ((time timestamp$1) near5 subtract$4) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:58
-	17	((transmi$6) near5 delay) same ((time$1 timestamp$1) near5 subtract$4) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:40
-	51	((transmi$6) near5 delay latency) same ((time$1 timestamp$1) near5 subtract$4) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:40
-	17	((transmi$6) near5 (delay latency)) same ((time$1 timestamp$1) near5 subtract$4) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:41
-	82	((up$1load$3 down$1load$3 up$1link down$1link) near5 (delay latency)) same (time timestamp$1) and g06f$.ipc.	USPAT; EPO	2003/11/10 15:43
-	155	((transmi$6) near5 delay) same ((time timestamp$1) near5 subtract$4)	USPAT; EPO	2003/11/10 16:24
-	4965	compar$4 near4 (time timestamp$1) with (delay latency)	USPAT; EPO	2003/11/10 16:24
-	488	((transmi$6) near5 delay) same ((time timestamp$1) near5 compar$4)	USPAT; EPO	2003/11/10 16:24
-	47	((transmi$6) near5 delay) same ((time timestamp$1) near5 compar$4) and g06f$.ipc.	USPAT; EPO	2003/11/10 16:25
-	25	((up$1load$3 down$1load$3 up$1link down$1link) with (RTD (round adj trip adj delay)))	USPAT; EPO	2003/11/10 16:46
-	1	((up$1load$3 down$1load$3 up$1link down$1link) with (RTD (round adj trip adj delay)))	JPO; IBM_TDB	2003/11/10 16:37
-	11	("3643031" | "3665405" | "3811013" | "3825899" | "3829777" | "3838221" | "3848093" | "3862370" | "3862373" | "3879580" | "3879581").PN.	USPAT	2003/11/10 16:41
-	28	4009345.URPN.	USPAT	2003/11/10 16:41
-	39	(("3643031" | "3665405" | "3811013" | "3825899" | "3829777" | "3838221" | "3848093" | "3862370" | "3862373" | "3879580" | "3879581").PN.) 4009345.URPN.	USPAT; EPO	2003/11/10 16:41
-	0	((("3643031" | "3665405" | "3811013" | "3825899" | "3829777" | "3838221" | "3848093" | "3862370" | "3862373" | "3879580" | "3879581").PN.) 4009345.URPN.) and RTD	USPAT; EPO	2003/11/10 16:41
-	6	((("3643031" | "3665405" | "3811013" | "3825899" | "3829777" | "3838221" | "3848093" | "3862370" | "3862373" | "3879580" | "3879581").PN.) 4009345.URPN.) and "round trip delay"	USPAT; EPO	2003/11/10 16:44
-	0	"09643916"	USPAT; EPO	2003/11/10 16:44
-	13	(((up$1load$3 up$1link) with (down$1load$3 down$1link)) with (RTD (round adj trip adj delay)))	USPAT; EPO	2003/11/10 16:47
-	554	((up$1load$3 down$1load$3 up$1link down$1link) with (sample$4))	USPAT; EPO	2003/11/12 15:30

-	86	((up$1load$3 down$1load$3 up$1link down$1link) with (sample$4)) and g06f$.ipc.	USPAT; EPO	2003/11/12 15:31
-	6	((up$1load$3 down$1load$3 up$1link down$1link) with ((sample$4) near5 time$1)) and g06f$.ipc.	USPAT; EPO	2003/11/12 15:32
-	52	((up$1load$3 down$1load$3 up$1link down$1link) with ((sample$4) near5 time$1))	USPAT; EPO	2003/11/12 15:33
-	5	(up$1link and down$1link) with (sample$4 near5 time$1)	USPAT; EPO	2003/11/12 15:34
-	1213	(up$1link and down$1link) with (time$1)	USPAT; EPO	2003/11/12 15:34
-	1260	(up$1link and down$1link) with (delay time$1)	USPAT; EPO	2003/11/12 15:35
-	343	(up$1link and down$1link) with (delay time$1) same (calculat$6 determin$4)	USPAT; EPO	2003/11/12 15:35
-	130	(up$1link and down$1link) with (delay time$1) with (calculat$6 determin$4)	USPAT; EPO	2003/11/12 15:36
-	8	(up$1link and down$1link) with (delay time$1) same (calculat$6 determin$4) same sampl$4	USPAT; EPO	2003/11/12 15:43
-	371	(sen$4 near7 receiv$4) with (uplink downlink)	USPAT; EPO	2003/11/12 16:58
-	24	(sen$4 near7 receiv$4) with (uplink downlink) with delay	USPAT; EPO	2003/11/12 16:58
-	2	(sen$4 near7 receiv$4) with (uplink downlink) same (RTD (round adj trip adj delay))	USPAT; EPO	2003/11/12 17:01
-	6	((sen$4 transmi$4) near7 receiv$4) with (uplink downlink) same (RTD (round adj trip adj delay))	USPAT; EPO	2003/11/13 08:35
-	812	(uplink downlink) near5 (latency delay time) with (sen$4 transmi$6 receiv$4)	USPAT; EPO	2003/11/13 08:38
-	175	(uplink downlink) near5 (latency delay time) with ((sen$4 transmi$6) near5 receiv$4)	USPAT; EPO	2003/11/13 08:55
-	1	(uplink downlink) near5 (latency delay time) with ((sen$4 transmi$6) near5 receiv$4) and g06f$.ipc.	USPAT; EPO	2003/11/13 08:39
-	6	(uplink downlink) near5 (latency delay) with (((sen$4 transmi$6) near5 receiv$4) near5 time$1)	USPAT; EPO	2003/11/13 09:01
-	2	(determin$4 calculat$4) near3 (average minimum maximum shortest longest) with ((latency delay) near5 (up$1link down$1link))	USPAT; EPO	2003/11/13 09:06
-	18	(average minimum maximum shortest longest) with ((latency delay) near5 (up$1link down$1link))	USPAT; EPO	2003/11/13 09:17
-	3	(minimum) with ((latency delay) near5 (up$1link down$1link))	USPAT; EPO	2003/11/13 09:17
-	9	(DC adj bias) same (RTD up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:53
-	192	(bias) same (RTD up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:53
-	73	(bias) with (RTD up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:53
-	4	(DC near2 bias) with (RTD up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:53
-	10	(DC near2 bias) same (RTD up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:54
-	1	((DC near2 bias) same (RTD up$1link$1 down$1link$1)) not ((DC adj bias) same (RTD up$1link$1 down$1link$1))	USPAT; EPO	2003/11/13 10:57
-	1	(wrap$3 adj around) with (up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 10:58
-	7	(wrap$3 adj around) same (up$1link$1 down$1link$1)	USPAT; EPO	2003/11/13 11:00
-	14	RTD with timer	USPAT; EPO	2003/11/13 13:04

-	4	RTD with time$1out	USPAT; EPO	2003/11/13 13:11
-	39	wireless with timeout	USPAT; EPO	2003/11/13 14:05
-	0	adjust$6 with (time near2 wraparound)	USPAT; EPO	2003/11/13 14:06
-	586	((wrap$3 adj around) wraparound) near5 time	USPAT; EPO	2003/11/13 14:06
-	1	((wrap$3 adj around) wraparound) near5 time with adjust$6	USPAT; EPO	2003/11/14 13:21
-	369	(roll$4over carry carry$1out overflow$4) near5 time with adjust$6	USPAT; EPO	2003/11/14 13:22
-	25	(roll$4over carry carry$1out overflow$4) near5 time with adjust$6 and g06f$.ipc.	USPAT; EPO	2003/11/14 13:25
-	13	(roll$4over carry carry$1out) near5 time with adjust$6 and g06f$.ipc.	USPAT; EPO	2003/11/14 13:24
-	0	(roll$4over) near5 time with adjust$6 and g06f$.ipc.	USPAT; EPO	2003/11/14 13:23
-	3	(roll$4over) near5 time with adjust$6	USPAT; EPO	2003/11/14 13:23
-	287	(roll$4over carry carry$1out) near5 time with adjust$6	USPAT; EPO	2003/11/14 13:24
-	3455	(roll$4over carry carry$1out overflow$4) near5 tim$4 and g06f$.ipc.	USPAT; EPO	2003/11/14 13:25
-	31	(roll$4over carry carry$1out overflow$4) near5 tim$4 same (uplink downlink)	USPAT; EPO	2003/11/14 13:29
-	2	(roll$4over overflow$4) near5 tim$4 same (uplink downlink)	USPAT; EPO	2003/11/14 14:06
-	96	local adj3 time with duration	USPAT; EPO	2003/11/14 14:06
-	1927904	time not counter	USPAT; EPO	2003/11/14 14:07
-	2230519	time not counter with wraparound	USPAT; EPO	2003/11/14 14:07
-	8	(time near2 wraparound) not counter	USPAT; EPO	2003/11/14 15:19
-	3	(time near2 wrap$1around) not counter	JPO; IBM TDB	2003/11/14 16:05

US006600731B2

(12) **United States Patent**
Menzel et al.

(10) **Patent No.:** **US 6,600,731 B2**
(45) **Date of Patent:** **Jul. 29, 2003**

(54) **METHOD AND BASE STATION SYSTEM FOR CALLING MOBILE STATIONS FOR THE TRANSMISSION OF PACKET DATA**

(75) Inventors: **Christian Menzel**, Maisach (DE); **Martin Öttl**, Weilheim (DE)

(73) Assignee: **Siemens Aktiengesellschaft**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/380,018**

(22) PCT Filed: **Jan. 26, 1998**

(86) PCT No.: **PCT/DE98/00229**

§ 371 (c)(1), (2), (4) Date: **Aug. 23, 1998**

(87) PCT Pub. No.: **WO98/38818**

PCT Pub. Date: **Sep. 3, 1998**

(65) **Prior Publication Data**

US 2003/0117990 A1 Jun. 26, 2003

(30) **Foreign Application Priority Data**

Feb. 24, 1997 (DE) 197 07 261

(51) **Int. Cl.**[7] **H04B 7/212**

(52) **U.S. Cl.** **370/347**; 370/442; 370/444; 370/468

(58) **Field of Search** 370/310, 321, 370/337, 347, 322, 348, 477, 498, 431, 437, 442, 443, 444, 336, 345, 468; 455/450, 452

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,404,355 A * 4/1995 Raith 370/347
5,729,541 A * 3/1998 Hamalainen et al. 370/337
6,031,832 A * 2/2000 Turina 370/348

FOREIGN PATENT DOCUMENTS

DE	44 02 903 A1	8/1995
DE	195 24 659 C1	10/1996
EP	0 687 078 A2	12/1995
WO	WO 94/05103	3/1994
WO	WO 95/12931	5/1995
WO	WO 95/16330	6/1995
WO	WO 95/32589	11/1995

OTHER PUBLICATIONS

Akesson, GPRS, General Packet Radio Service, IEEE, pp. 640–643, 1995.*

Scourias, Overview of GSM: The Global System for Mobile Communications, University of Waterloo, pp. 1–26.*

Brasche, G., "Evaluation of a MAC Protocol Proposed for a General Packet Radio service in GSM", IEEE Symposium on Personal, Indoor & Mobile Radio Communications, vol. 2, (1996), pp. 668–672.

* cited by examiner

Primary Examiner—Dang Ton
Assistant Examiner—Frank Duong
(74) *Attorney, Agent, or Firm*—Staas & Halsey LLP

(57) **ABSTRACT**

In order to call mobile'stations, mobile stations are, according to the invention, allocated to time slots for calling on the basis of subscriber-specific profiles. The number of time slots allocated for calling within a macroframe can be set individually for the mobile, stations. The requirement for time slots allocated for calling is thus covered individually for each mobile station. For time-critical applications, shorter delay times can be achieved using the method according to the invention.

18 Claims, 3 Drawing Sheets

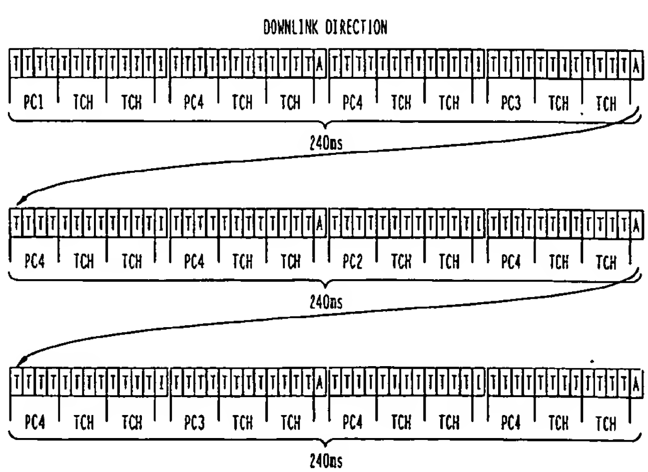


FIG. 1

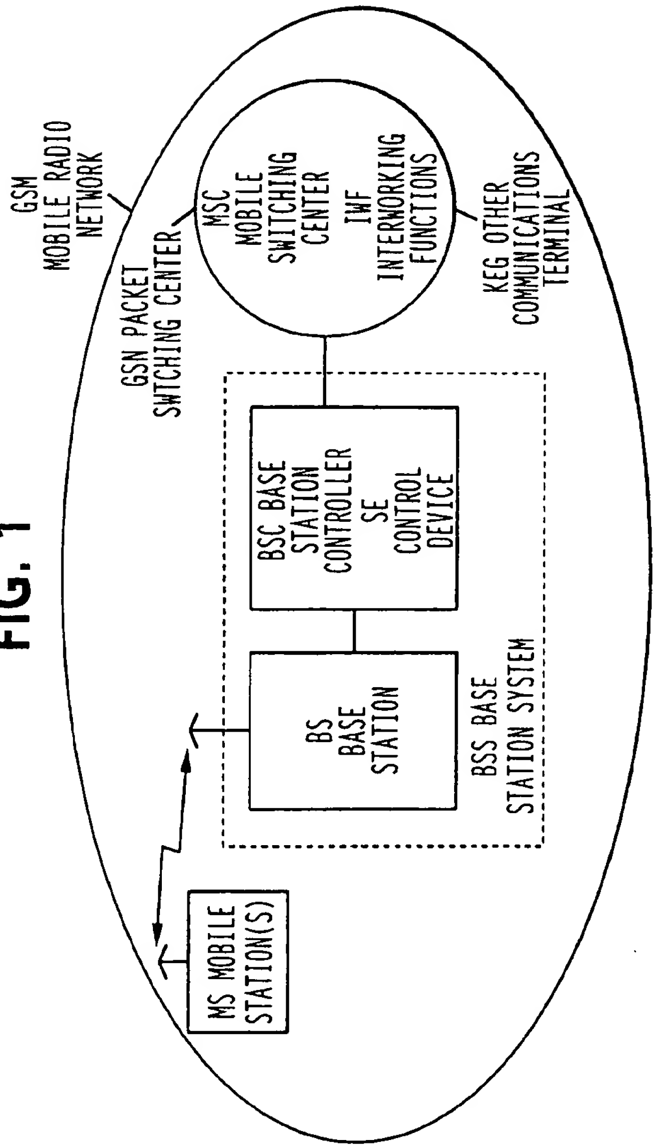

GSM
MOBILE RADIO
NETWORK
GSN PACKET
SWTCHING CENTER
MSC
MOBILE
SWITCHING
CENTER
IWF
INTERWORKING
FUNCTIONS
KEG OTHER
COMMUNICATIONS
TERMINAL
BSC BASE
STATION
CONTROLLER
SE
CONTROL
DEVICE
BS
BASE
STATION
BSS BASE
STATION SYSTEM
MS MOBILE
STATION(S)

FIG. 2

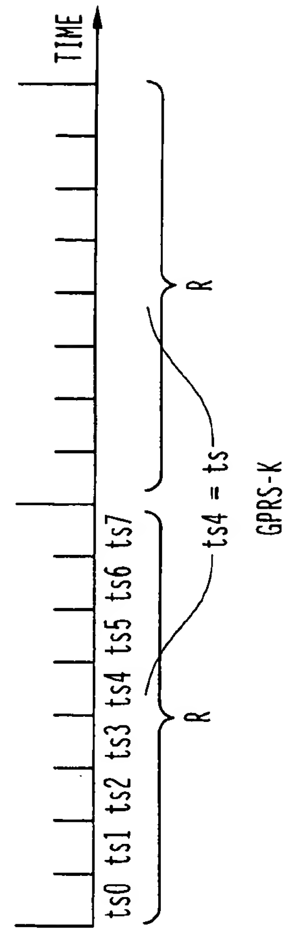

TIME
ts0 ts1 ts2 ts3 ts4 ts5 ts6 ts7
R
R
ts4 = ts
GPRS-K

FIG. 3

DOWNLINK DIRECTION

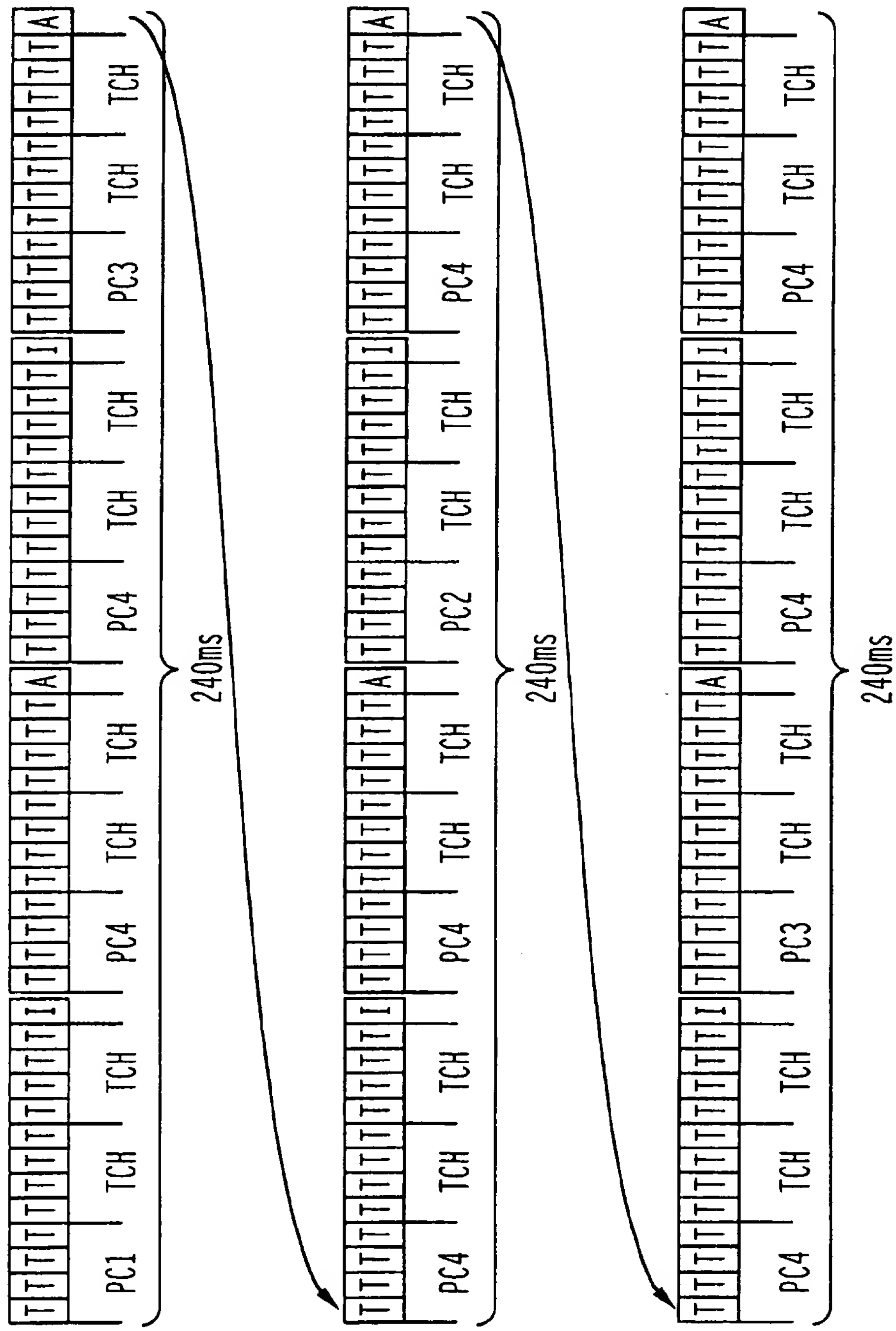
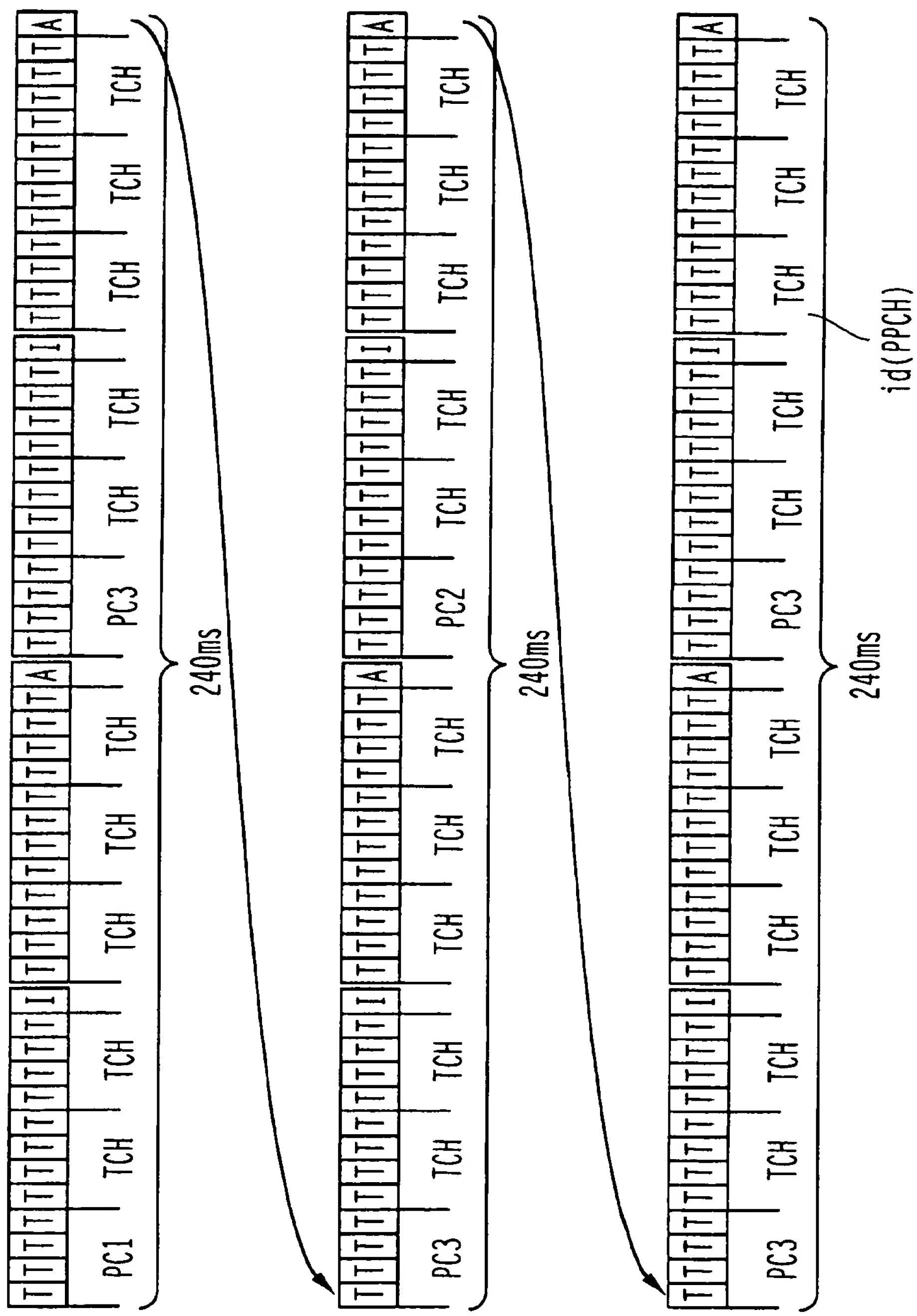


FIG. 4

DOWNLINK DIRECTION



METHOD AND BASE STATION SYSTEM FOR CALLING MOBILE STATIONS FOR THE TRANSMISSION OF PACKET DATA

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to packet data transmissions between mobile and base radio stations, and more specifically to a method and system for calling mobile stations in a mobile radio network.

2. Description of the Prior Art

Connection-oriented concepts and concepts based on logic links may be used for transmitting data between two communications terminals. In the case of connection-oriented data transmissions, physical resources must be provided between the two communications terminals throughout the entire time of data transmission.

When transmitting data via logic links, there is no need for any permanent provision of physical resources. One example of such a data transmission is packet data transmission. In this case, there is a logic link between the two communications terminals throughout the duration of the overall data transmission, but physical resources are provided only during the times when data packets are actually being transmitted. This method is based on the data being transmitted in short data packets, between which relatively long pauses may occur. The physical resources are available for other logic links in the pauses between the data packets. The use of a logic link saves physical resources.

The packet data transmission method which is known from German Patent DE 44 02 930 may be used in particular for communications systems with limited physical resources. For example, in mobile radio systems such as the GSM mobile radio system (Global System for Mobile Communications), the physical resources in the frequency domain—the number of frequency channels and time slots—are limited and must be used economically.

The GSM mobile radio system is one example of a time-division multiplex mobile radio system, in which time slots within a frequency channel may be split between different communications terminals. The radio station at the network end of a mobile radio network is a base station, which communicates with mobile stations via a radio interface. Transmission from a mobile station to the base station is called the uplink direction, and transmission from the base station to a mobile station is called the downlink direction. A channel, which is reserved for packet,data transmission, is formed by at least one time slot per time-division multiplex frame. A plurality of time-division multiplex frames in this case form a macroframe. Furthermore, the channel is governed by the carrier frequency and, possibly, by a frequency jump sequence.

The GSM mobile radio system was originally designed for voice transmission, although one channel was reserved for continuous information transmission between the mobile station and base station. However, for packet data transmission, a common packet data channel is used for packet data transmission for a plurality of mobile stations.

If one wants to transmit data from the network to the mobile station, that is to say, in the downlink direction, this mobile station is called by the network via the base station system (paging). To do this, time slots for calling are provided within the packet data channel, time slots for signalling messages, are required to set up a call, but no useful data are transmitted.

In this scheme the time interval between two successive call possibilities is the same for all mobile stations. But the waiting time between two successive call possibilities is too long for applications which require immediate data transmission.

SUMMARY OF THE INVENTION

In consequence, the invention is based on the object of specifying an improved method and an improved base station system for calling mobile stations in a mobile radio network. This object is achieved by the method having the features of forming a packet data channel by at least one time slot per time-division multiplex frame; forming a macroframe from a plurality of time-division multiplex frames; carrying out said packet data transmission from a plurality of said mobile stations via a common said packet data channel; providing at least one time slot for calling said mobile stations at cyclical intervals in said packet data channel; and setting a number of said time slots allocated within said macroframe for said mobile stations individually, where said.mobile stations are allocated to time slots for calling on the basis of subscriber-specific profiles and by the base station system having the features of a packet data channel that is formed by at least one time slot per time division multiplex frame; a macroframe that is formed by a plurality of time-division multiplex frames; a packet data transmission from a plurality of said mobile stations that are carried out via a common said packet data channel; at least one time slot for calling said mobile stations that is provided at cyclic intervals in said packet data channel; a control device for calling said mobile stations, said control device allocating said mobile stations to time slots for calling on the basis of subscriber-specific profiles, wherein a number of said time slots allocated within said macroframe for said mobile stations is set individually.

According to the invention, in the case of the method for calling mobile stations, mobile stations are allocated to time slots for calling on the basis of subscriber-specific profiles. The number of time slots allocated for calling within a macroframe can be set individually for the mobile stations.

This allows the requirement for time slots allocated for calling to be covered individually for each mobile station. For time-critical applications, shorter delay times can be achieved using the method according to the present invention. The number of time slots allocated for calling is governed on the basis of subscriber-specific profiles which are stored at the network end and/or at the mobile station end, and are signalled when required between the mobile station and the base station system.

According to an advantageous refinement of the invention, a plurality of groups are formed for calling mobile stations. A mobile station is allocated to one of these groups, and the groups can be allocated different numbers of time slots for calling within one cycle. The signalling complexity to form such groups is low, and group formation therefore allows the individual requirements of the mobile stations to be met quickly. Furthermore, it is advantageous if a mobile station is allocated to a plurality of groups at the same time. This allows a reduced delay time when calling to be achieved very easily for this mobile station.

According to an advantageous development of the invention, the association with the groups is signalled to the mobile stations by the base station. In this way, a changed requirement from the mobile station is satisfied quickly, flexibly and with only a small amount of signalling complexity.

The changed requirement is advantageously signalled by the mobile station to the base station system by changing its subscriber-specific profile, and the base station system then changes the mobile station's allocation of time slots for calling. Subscriber-specific profiles, which may also be used for other resource allocations in a subsequent useful channel allocation, are particularly suitable as a criterion for the requirement for time slots for calling. Each mobile station may be allocated a specific profile independently of other mobile stations, and, in the process, the resource allocation by the base station system can be optimized, taking into account all of the mobile stations to be supplied.

In order to reduce the delay times for the mobile stations further, time slots for useful data transmission are also used as time slots for calling. One or more mobile stations may additionally be called via these time slots. The interval between two time slots for calling is thus shortened. Abbreviated identifiers which are transmitted in the time slots for useful data transmission expediently indicate the presence of calls for the mobile stations.

For mobile stations in which the time delays for a call are not critical, a long interval (possibly a number of seconds) may be set by a suitable allocation of time slots for calling. A mobile station can thus be switched between the time slots allocated to it for calling, in an energy-saving mode, reducing energy consumption of this mobile station and allowing relatively long operating times before the battery needs to be recharged. However, the application of the method according to the invention is not just limited to mobile called radio stations. The called mobile station may also be stationary.

According to advantageous further developments of the invention, the subscriber-specific profiles are designated by different qualities of service. These qualities of service are, for example, for the GSM mobile radio network, standardized identifiers of a packet data transmission which can also easily be used for the method according to the invention. As an alternative to this, the subscriber-specific profiles are designated by different services used by the mobile station. The delay which is still acceptable or even desirable can easily be determined as well on the basis of the service to be used, for example E-mail (relatively long delay) or electronic train control (short delay). In addition, it is possible to base the allocation of time slots for calling on the number of mobile stations registered in the radio area of the base station system. This knowledge can be included in an improved allocation strategy.

Further advantageous refinements of the subscriber-specific profiles are designated by different priorities, by different permissible delay times or by different required data rates for packet data transmission. The permissible intervals between two time slots allocated to the respective mobile station for calling may be read directly from these values.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be explained in more detail in the following text with reference to exemplary embodiments and by using illustrative drawings, in which:

FIG. 1 is a block diagram of a time-division multiplex mobile radio system for packet data transmission,

FIG. 2 is a diagram showing a frequency channel using time-division multiplex,

FIG. 3 is a diagram showing the time slots in a channel for a packet data transmission and for a call from mobile stations without group formation, and

FIG. 4 is a diagram showing the time slots in a channel for a packet data transmission and for a call from mobile stations with group formation.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The time-division multiplex mobile radio system according to FIG. 1 is, for example, a GSM mobile radio network GSM which contains at least one base station system BSS having a base station controller BSC and a base station BS. Mobile stations MS are located in the radio area of the one base station BS illustrated. The base station system BSS produces the link to other devices in the GSM mobile radio network GSM. The base station controller BSC contains a control device SE which allocates radio resources for the mobile stations MS. However, the control device SE may also be contained in other devices in the mobile radio system.

These other devices are, for example, a mobile switching center MSC and a unit for providing inter working functions IWF. The interaction of a mobile switching center MSC and interworking functions IWF results in a packet switching center, which is also called a GSN (GPRS support node). This packet switching center is connected to an MSC for voice switching, but alternatively could be a remote, dedicated unit.

The GSM mobile radio network GSM may be connected to other communications networks. By way of example, another communications terminal KEG can be connected to the GSM mobile radio network, or may even be a component of this GSM mobile radio network GSM.

The GSM mobile radio network GSM is intended to be used for packet data transmission in parallel with the known voice transmission. In this case, the device for producing interworking functions IWF may produce the coupling of the GSM mobile radio network GSM to data transmission networks and thus for the other communications terminal KEG.

The radio interface between the mobile stations MS and a base station BS is characterized by a frequency and at least one time slot ts. According to FIG. 2, for example, eight time: slots ts (ts0 to ts7) are combined to form a frame R. The frame R is repeated cyclically, a recurring time slot, for example the time slot ts=ts4, belonging to one channel. This time slot ts is used in the following text as the channel GPRS-K for packet data transmission for the purpose of the GPRS (General Packet Radio Services) service. A plurality of time-division multiplex frames R may be combined to form a macroframe.

If a mobile station MS intends to use this service, then, in accordance with the GSM terminology, it makes a random access using a short, so-called access burst, and changes to a dedicated control channel. This is followed by authentication and setting of the context for a logic link (standby state). If the other communications terminal KEG intends to communicate with a mobile station MS via the packet data service, the desired mobile station MS is called (paging) and the indicated random access takes place, at the network end.

For another packet data transmission in the downlink direction, the mobile station MS is allocated an abbreviated MS identifier and the corresponding GPRS channel GPRS-K. The timing advance and the reception level in the base station BS are then defined at the network end. Four successive time slots T are then transmitted as a packet data block TCH in the downlink direction to the mobile station MS designated by the abbreviated MS identifier.

The calling of a mobile station MS is illustrated with reference to FIGS. 3 and 4, three macroframes in each case being combined to form a higher-order frame.

Four time slots T for packet data transmission are in each case combined with time slots PPCH for calling to form a packet data block TCH or a block PC1, PC2, PC3, PC4. Three such packet data blocks TCH and one time slot A, or I for signalling are repeated four times to form a macroframe, which comprises 52 frames R. A macroframe lasts for 240 ms.

The information in a packet data block TCH is interleaved with four time slots T. The allocation of packet data blocks TCH to different mobile stations MS is carried out flexibly in the uplink and downlink directions to one or more mobile stations MS. Different data rates may therefore be used. Priorities may be used to distinguish between the mobile stations MS for access to the GPRS channel. The application of packet data blocks TCH while a logic link is in existence takes place in the band, that is to say, within the packet data blocks TCH, indicator messages are used to indicate to the mobile stations MS which of them may use the following packet data blocks TCH.

four successive time slots T for packet data transmission are interleaved in the downlink direction. The intermediate time slots I are used for measurements relating to the mobile stations MS in adjacent cells, and the time slots A are used for signalling. The sequence of time slots A, I for signalling and adjacent channel measurement may be based on different sequences, for example A/I=1/1.

According to a first exemplary embodiment, all the mobile stations MS are allocated to a common group for calling mobile stations MS. The blocks PC1, PC2, PC3 and PC4 with time slots PCCH for calling are used for calling.

These blocks PC1, PC2, PC3 and PC4 are allocated on the basis of subscriber-specific profiles for the mobile stations MS. The subscriber-specific profiles are formed by quality of service classes QoS **1** to **4**, the average delay times (transfer delay), which comprise a delay in the uplink and downlink directions, and the data rates that can be transmitted respectively being defined for the quality of service classes QoS. A quality of service class QoS is defined and stored for the mobile stations MS. The quality of service classes QoS may be changed by appropriate signalling by the mobile station MS or by network presets. As an alternative to the quality of service classes QoS, other criteria, which have already been mentioned above, may be used for the subscriber-specific profiles.

According to FIG. **3**., the blocks PC1, PC2 and PC3 are allocated to mobile stations MS using the quality of service class QoS **2**, so that a call is guaranteed within 180 ms. The block PC4 is also allocated to the mobile stations using the quality of service class QoS **1**, so that the interval between two calls is reduced by a factor of three to 60 ms for these mobile stations MS. The delay before a data packet is transmitted is thus also reduced for these mobile stations MS. Only the block PC1 is allocated to mobile stations MS for the quality of service class QoS **4**, that is to say the lowest priority, the interval in this case is 720 ms. The mobile stations MS for which the interval is 720 ms may, in particular, switch to an energy-saving mode for this time period, providing a considerable energy saving in comparison with more frequent calling. Should one of these mobile stations MS using the quality of service class QoS **4** change to the next higher quality of service class QoS **3**, the block PC2 may be allocated in addition. If the base station does not require the blocks PC2, PC3 and PC4 for calling mobile stations, they may be used for useful data transmission.

According to a second exemplary embodiment for calling mobile stations MS, the mobile stations MS are allocated to two groups **1** or **2**. The blocks PC1, PC2, PC3 and PC4 with time slots PCCH for calling are once again used for calling.

According to FIG. **4**, the block PC1, which is repeated only every 720 ms, is allocated to the mobile stations MS using the quality of service class QoS **4** in the first group **1**. The block PC2 is allocated to the mobile stations MS using the quality of service class QoS **4** in the second group **2**. Calling occurs rarely for these mobile stations MS.

Mobile stations MS using the quality of service class QoS **3** in both groups **1** and **2** are called in the blocks PC1 and PC2, which results in the interval between two possible calls being halved. The blocks PC1, PC2 and PC3 are used for calling for the mobile stations MS using the quality of service classes QoS **1** and **2**, which results in a minimum interval of only 120 ms for these mobile stations MS. If the base station does not require the block PC3 for calling mobile stations, it may be used for useful data transmission.

If it is intended to reduce the interval, and thus the delay, further, then a packet data block TCH for calling may be used. In this case, it may be advantageous for a special abbreviated identifier id (the abbreviated identifiers id are normally used for allocation of packet data blocks TCH to mobile stations MS) to identify the that this block is used for calling. This abbreviated identifier, is signalled in advance to specific mobile stations BS.

Particularly suitable applications include packet-oriented transmission of information via the radio interface for telematics applications, fax and file transmission, point of sales implementations, fleet management and traffic management systems.

I claim:

1. A method for calling mobile stations for a packet data transmission in a time-division multiplex mobile radio system, comprising:

forming a packet data channel by at least one time slot per time-division multiplex frame;

forming a macroframe from a plurality of time-division multiplex frames;

carrying out said packet data transmission from a plurality of said mobile stations via a common said packet data channel;

providing at least one time slot for calling said mobile stations at cyclical intervals in said packet data channel; and

setting a number of time slots allocated for calling within said macroframe for said mobile stations individually, where said mobile stations are allocated to said time slots for calling on the basis of subscriber-specific profiles.

2. A method according to claim **1** which further comprises forming a plurality of groups for calling said mobile stations.

3. A method according to claim **2** wherein said forming a plurality of groups further comprises allocating a mobile staton to said plurality of groups at the same time.

4. A method according to claim **2** wherein said forming a plurality of groups further comprises signaling associations with said groups to said mobile stations by a base station.

5. A method according to claim **1** wherein said setting the number of said time slots allocated within said macroframe further comprises:

signaling to a base station by a mobile station a change to the mobile station's subscriber-specific profile; and

changing the allocation of said time slots for said mobile station.

6. A method according to claim **1** further comprising of using time slots for useful data transmission for said time slots for calling.

7. A method according to claim 6, wherein said using time slots for useful data transmission for said time slots for calling further comprises the step of indicating the presence of calls for said mobile stations by transmitting abbreviated identifiers in said time slots for useful data transmissions.

8. A method according to claim 1, further comprising switching a mobile station to an energy-saving mode between said allocated time slots for calling.

9. A method according to claim 1 further comprising designating said subscriber-specific profiles by different quality of service classes.

10. A method according to claim 1 further comprising, designating a subscriber-specific profiles by different services used by a mobile station.

11. A method according to claim 1 further comprising designating said subscriber-specific profiles by different priorities.

12. A method according to claim 1 further comprising designating said subscriber-specific profiles by different permissible delay times.

13. A method according to claim 1 further comprising designating said subscriber-specific profiles by different required data rates for packet data transmission.

14. A method as recited in claim 1, wherein a higher priority of a subscriber specific profile results in a higher number of respective time slots set.

15. A base station system for a packet data transmission to and from mobile stations in a time-division multiplex mobile radio system comprising a control device for controlling a radio interface between said base station system and said mobile stations, wherein said radio interface comprises:

a packet data channel that is formed by at least one time slot per time division multiplex frame;

a macroframe that is formed by a plurality of time-division multiplex frames;

a packet data transmission from a plurality of said mobile stations that are carried out via a common said packet data channel;

at least one time slot for calling said mobile stations that is provided at cyclic intervals in said packet data channel;

a control device for calling said mobile stations, said control device allocating said mobile stations to time slots for calling on the basis of subscriber-specific profiles, wherein a number of time slots allocated for calling within said macroframe for said mobile stations is set individually.

16. A system as recited in claim 15, wherein a higher priority of a subscriber specific profile results in a higher number of respective time slots set.

17. A computer readable storage for calling mobile stations for a packet data transmission in a time-division multiplex mobile radio system, controlling a computer by:

forming a packet data channel by at least one time slot per time-division multiplex frame;

forming a macroframe from a plurality of time-division multiplex frames;

carrying out said packet data transmission from a plurality of said mobile stations via a common said packet data channel;

providing at least one time slot for calling said mobile stations at cyclical intervals in said packet data channel; and

setting a number of time slots allocated for calling within said macroframe for said mobile stations individually, where said mobile stations are allocated to time slots for calling on a basis of subscriber-specific profiles.

18. A computer readable storage as recited in claim 17, wherein a higher priority of a subscriber specific profile results in a higher number of respective time slots set.

* * * * *

US005958060A

United States Patent [19]

Premerlani

[11] **Patent Number: 5,958,060**

[45] **Date of Patent: Sep. 28, 1999**

[54] **METHOD AND APPARATUS FOR CLOCK CONTROL AND SYNCHRONIZATION**

[75] Inventor: **William James Premerlani**, Scotia, N.Y.

[73] Assignee: **General Electric Company**, Schenectady, N.Y.

[21] Appl. No.: **09/002,432**

[22] Filed: **Jan. 2, 1998**

[51] **Int. Cl.**[6] **G06F 1/04**

[52] **U.S. Cl.** **713/400**; 713/501; 713/600

[58] **Field of Search** 395/551, 552, 395/553, 558, 555, 556; 377/43, 47, 48, 50

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,806,918	4/1974	Cauthron et al.	340/347 DD
4,715,000	12/1987	Premerlani	364/484
5,514,978	5/1996	Koegl et al.	324/772
5,576,664	11/1996	Herold et al.	331/1 A
5,598,448	1/1997	Girardeau, Jr.	375/376
5,742,649	4/1998	Muntz et al.	375/371
5,826,066	10/1998	Jardine et al.	395/551

FOREIGN PATENT DOCUMENTS

9612969 5/1996 WIPO .

OTHER PUBLICATIONS

"Internet Time Synchronization: The Network Time Protocol" by DL Mills, IEEE Trans on Communications, vol. 39, No. 10, Oct. 1991.

"Charge Comparison Protection of Transmission Lines–Relaying Concepts" by Leonard J. Ernst, Et Al, 1992 IEEE, 92 WM. 209-7 PWRD, pp. 1–9.

"Charge Comparison Protection of Transmission Lines–Communications Concepts" by Norman P. Albrecht, et al, 1992 IEEE, 92 WM 210-5 PWRD, pp. 1-7.

"High Speed Phase Segregated Line Differential Relay" by H. Peterson, ABB Inc., Canada, Paper #102, Mar. 1995 Vancouver, B.C. pp. 1–10.

"Type HCD PCM Current Differential Carrier Relay Scheme" by Mitshubishi Electric Corp., 15 Pages.

"Digital Current Differential Relay Type D2L7E" by Toshiba, pp. 1-11.

"Synchronized Phasor Measurements in Power System" by AG Phadke, IEEE Computer Applications in Power, Apr. 1993, pp. 10–15.

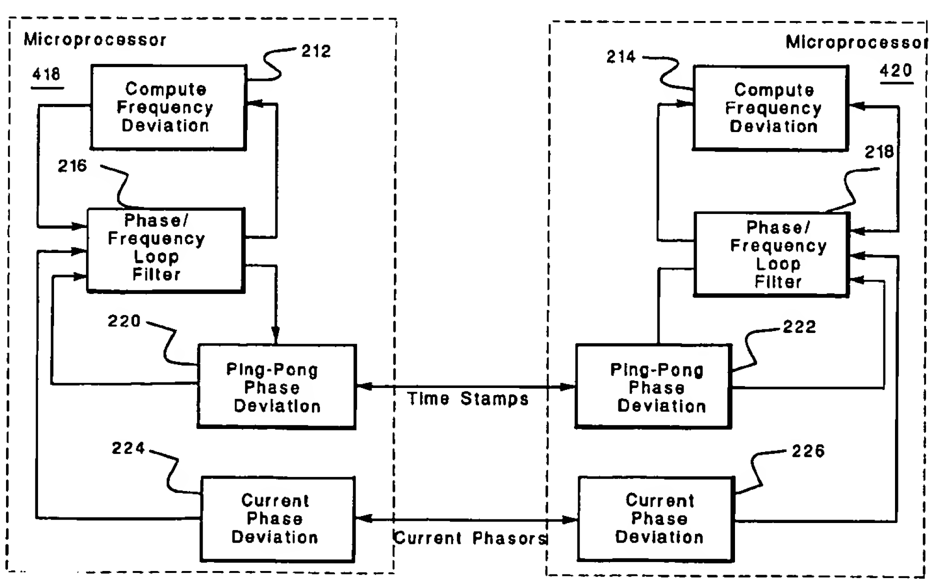
"Computer Relaying for Power Systems" By Arun Phadke, et al, 1988, Chap 3.6, Discrete Fourier Transform–pp. 85–88; Chap 4.3 Relaying as Parameter Estimation–pp. 123–129; Chap 5.2 Power Transformer Algorithms–pp. 166–176; Chap 8.9 Adaptive Relaying pp. 260–263.

Primary Examiner—Dennis M. Butler
Attorney, Agent, or Firm—Ann M. Agosti; Jill M. Breedlove

[57] **ABSTRACT**

A method for synchronizing clocks includes: sensing currents at multiple terminals; exchanging current and time stamp data between local and remote terminals; estimating a frequency deviation between local clock and power system frequencies; estimating a time based phase deviation with the time stamp data; estimating a current based phase deviation between the currents at the local and remote terminals; and using the frequency, time based phase, and current based phase deviations to synchronize the local clock. An integer counter value of the clock can be controlled by adjusting the integer counter value based on a sum of fractional counter values to increase clock resolution.

10 Claims, 5 Drawing Sheets



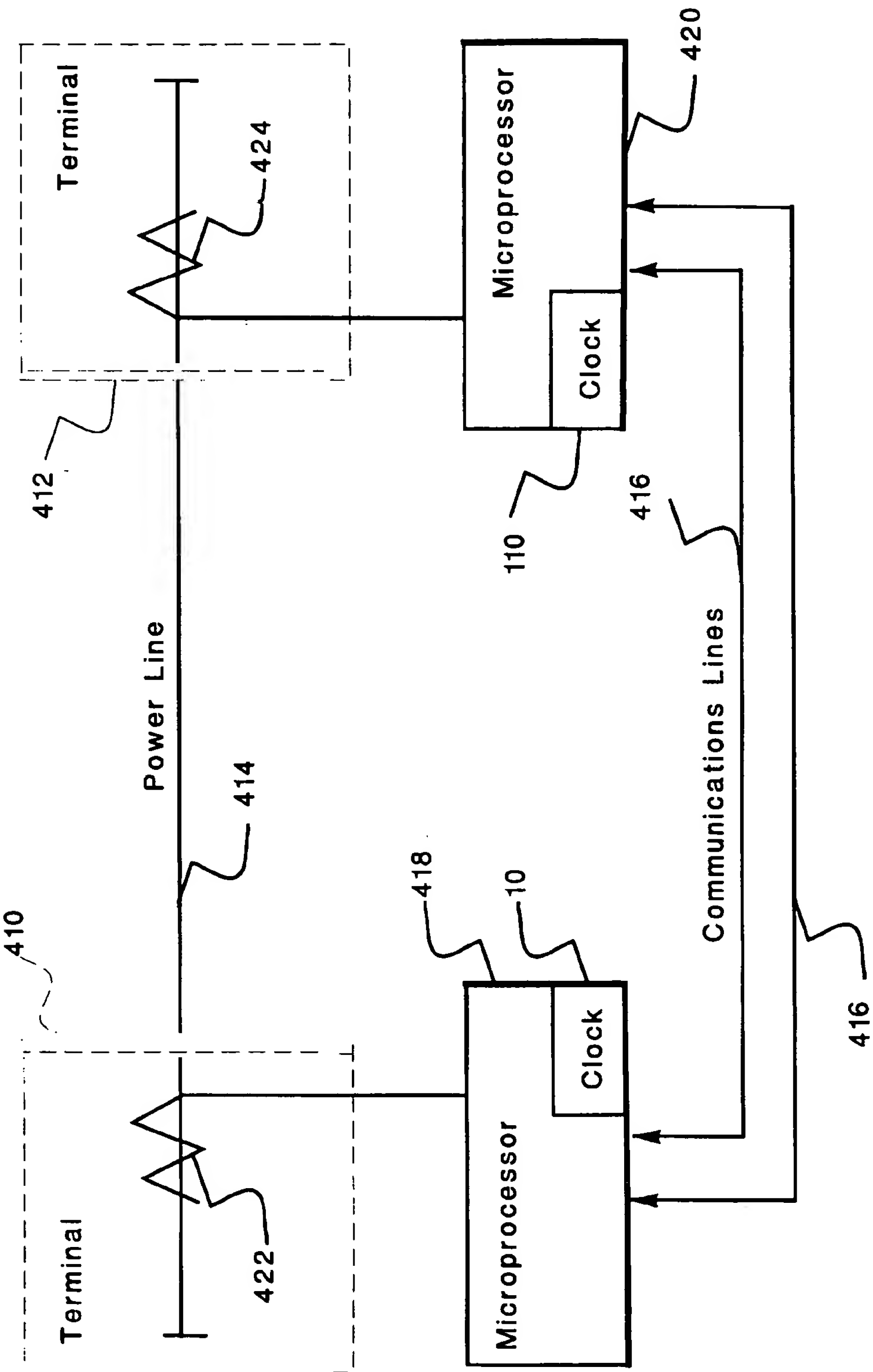

Terminal
424
412
Microprocessor
Clock
420
110
416
Power Line
414
Communications Lines
410
418
10
Terminal
422
Microprocessor
Clock
416

FIG. 1

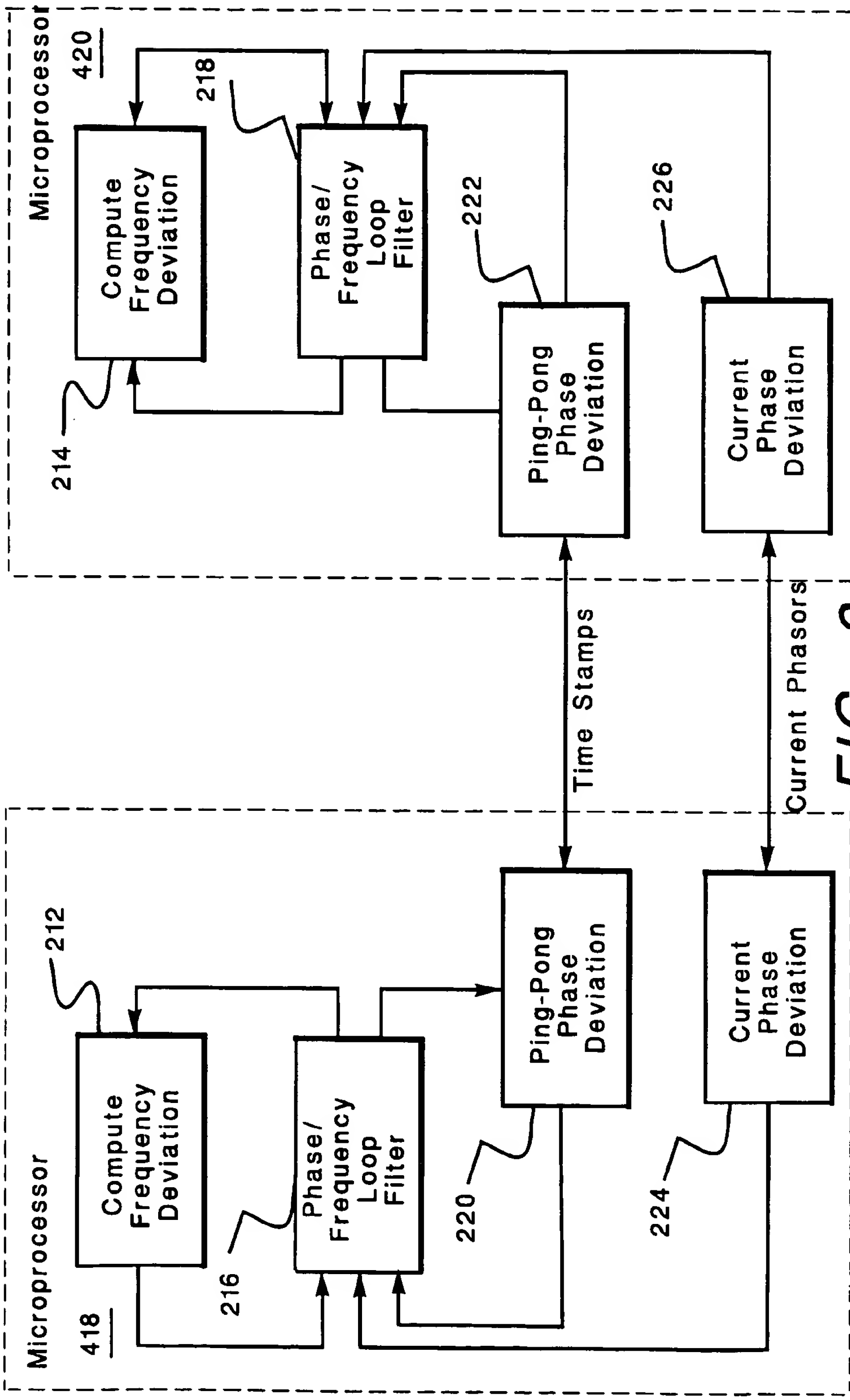

Microprocessor
420
Compute Frequency Deviation
214
Phase/ Frequency Loop Filter
218
Ping-Pong Phase Deviation
222
Current Phase Deviation
226
Time Stamps
Current Phasors
Microprocessor
418
Compute Frequency Deviation
212
Phase/ Frequency Loop Filter
216
Ping-Pong Phase Deviation
220
Current Phase Deviation
224

FIG. 2

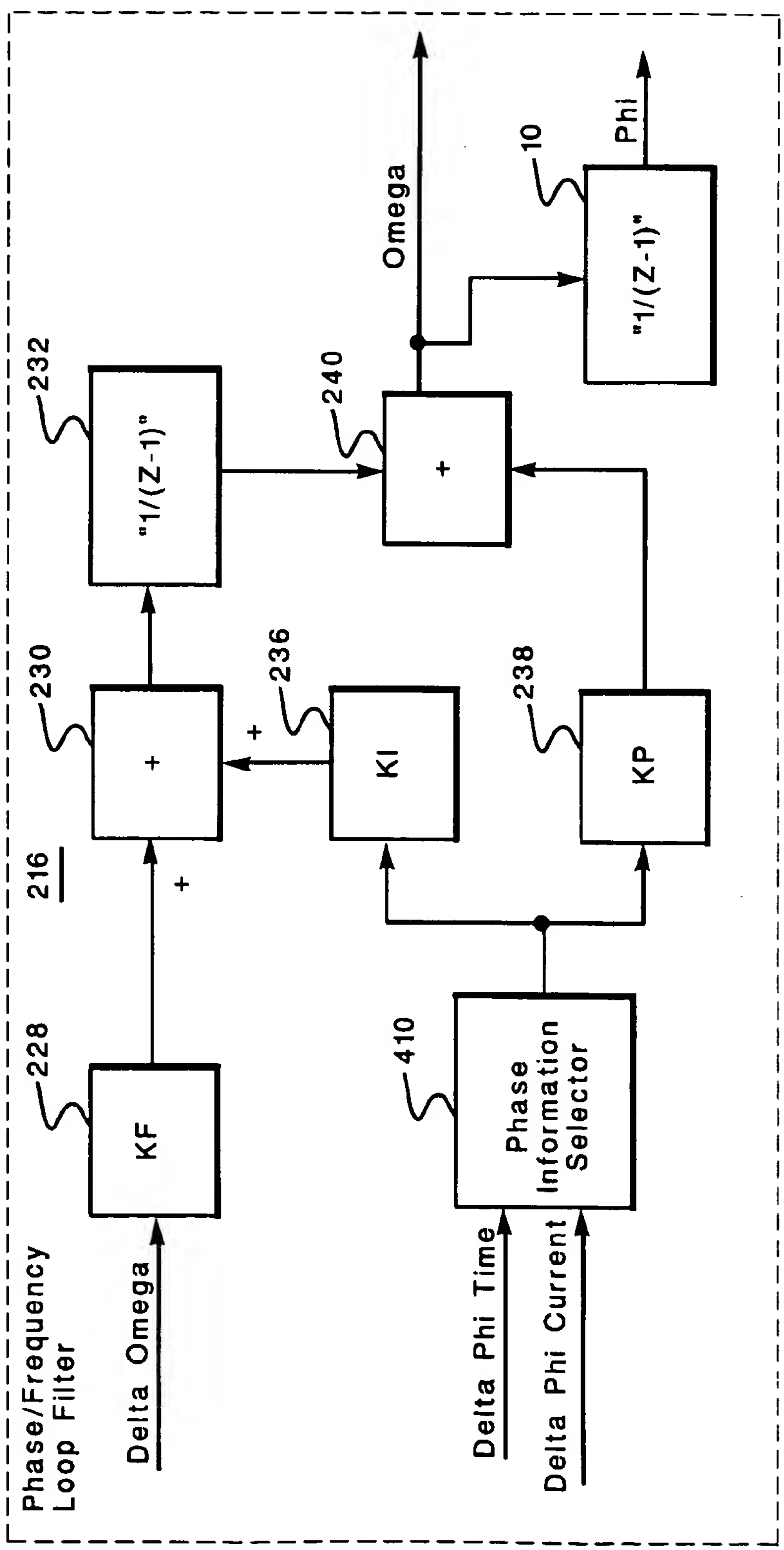

Phase/Frequency
Loop Filter
216
Delta Omega
KF
228
+
230
+
+
236
KI
"1/(Z-1)"
232
240
+
Omega
238
KP
Phase
Information
Selector
410
Delta Phi Time
Delta Phi Current
"1/(Z-1)"
10
Phi

FIG. 3

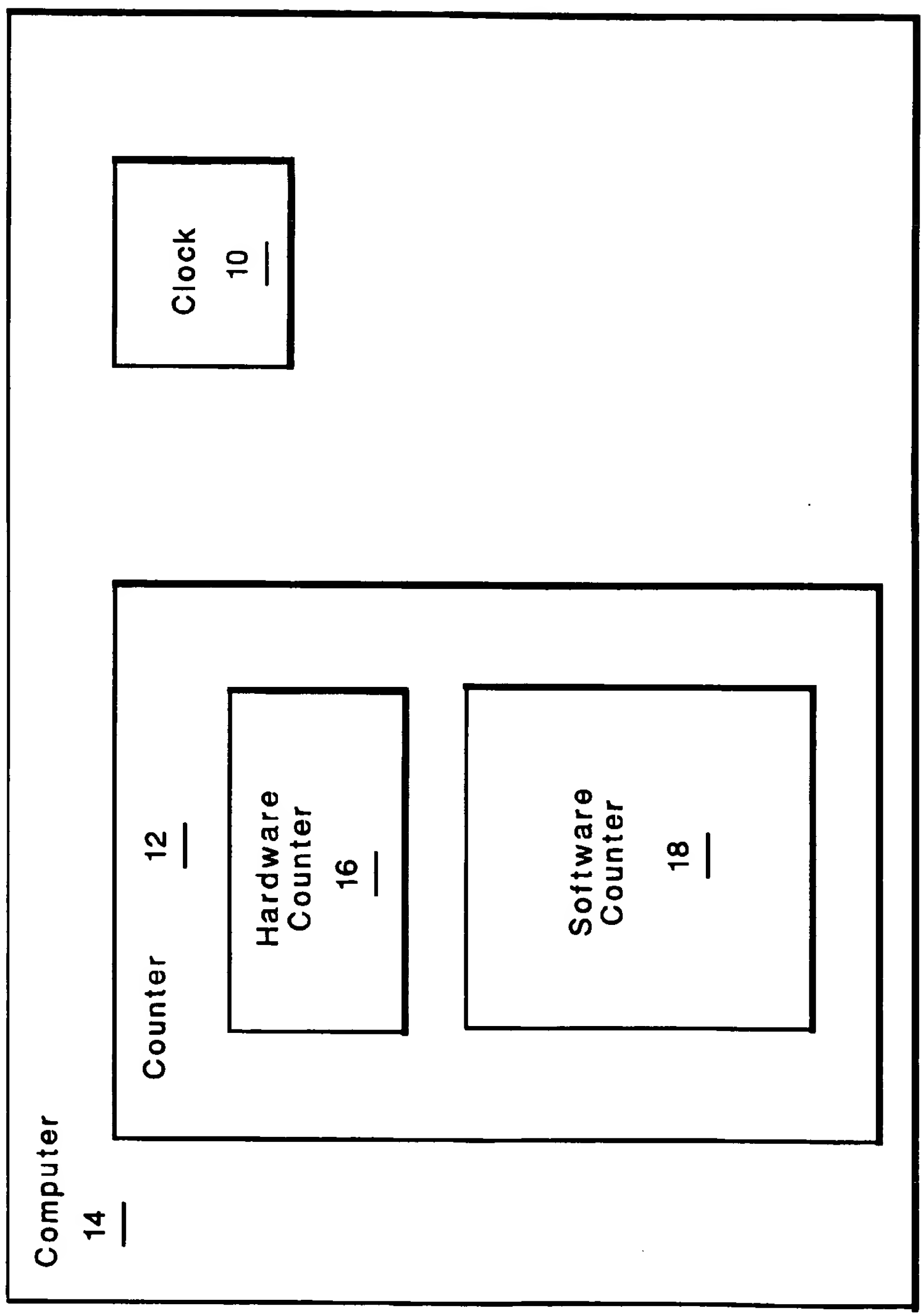

Clock
10
Computer
14
Counter
12
Hardware
Counter
16
Software
Counter
18

FIG. 4

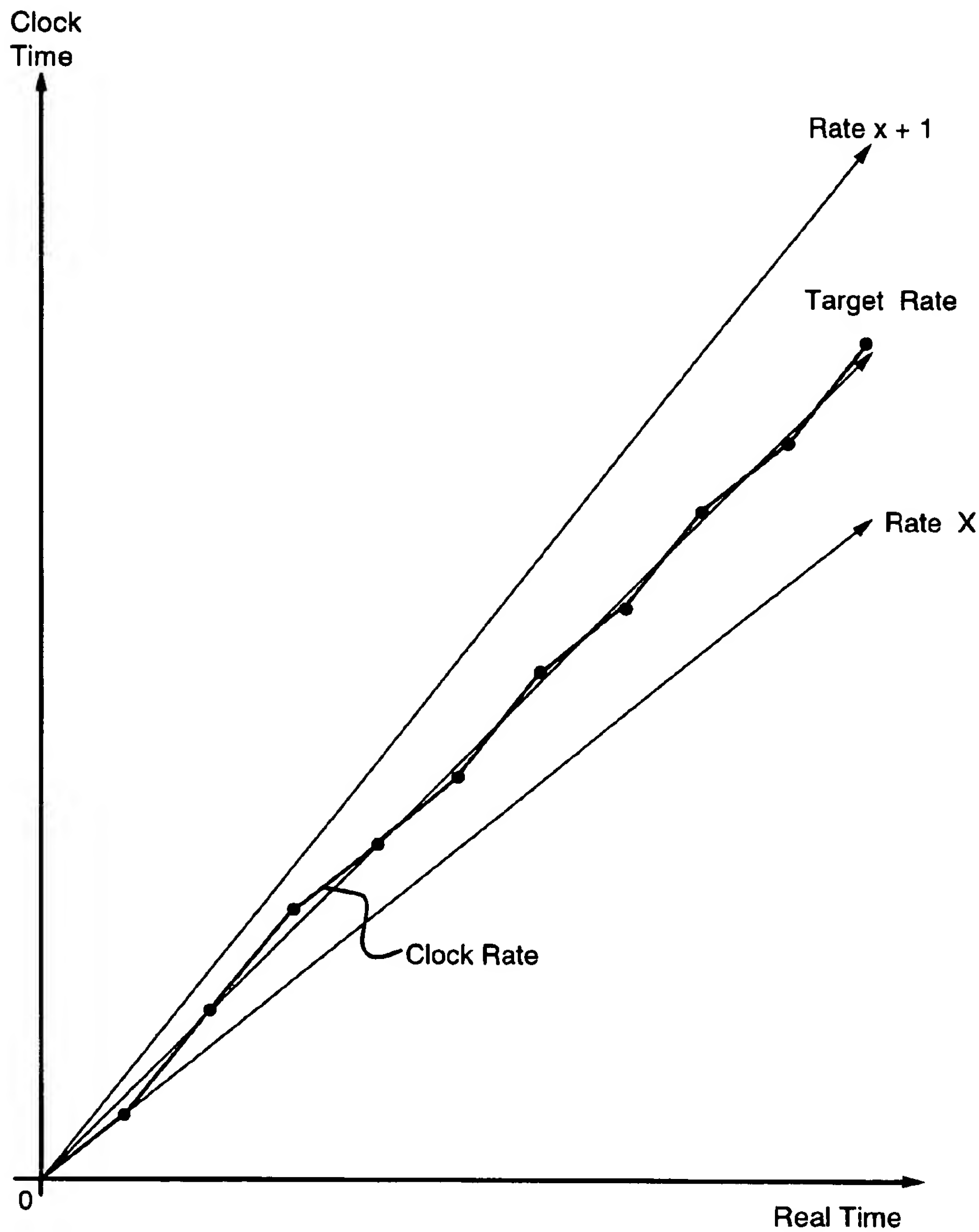


FIG. 5

METHOD AND APPARATUS FOR CLOCK CONTROL AND SYNCHRONIZATION

BACKGROUND OF THE INVENTION

The present invention relates to the control and synchronization of clocks used for driving counters to generate control pulses.

Conventional methods include driving a digital counter with a crystal-controlled clock. The counter is typically initialized with a value dependent on the base crystal frequency and the desired pulse rate. The counter then counts down by one each time it receives a pulse from the crystal. When the counter reaches zero, it produces an output pulse, resets to the initial count, and starts a new count down cycle. The resolution in the output rate of conventional counters is limited by the pulse frequency of the crystal. For example, a ten (10) megahertz crystal produces a resolution of 0.1 microseconds. For an application requiring a controllable output pulse rate of one (1) pulse every 250 microseconds, the resolution is 1 part in 2500. Some applications, however, such as applications that require close synchronization of data sampling at multiple locations, require a higher resolution.

Many power system monitoring, protection, and control functions could be performed more efficiently and accurately if power system digital measurements at multiple locations were synchronized. Generally such measurements are only somewhat synchronized because of difficulty in accurately synchronizing sampling clocks physically separated by large distances. Conventional uses of digital communications to synchronize sampling clocks at remote locations have accuracies limited by uncertainties in the message delivery time. In particular, digital communications can have different delays in different directions between a pair of locations which lead to an error in clock synchronization.

In addition to being important for multi-terminal transmission lines, time synchronization is important in many other applications such as power relays, determinations of sequences of events, economic power dispatch, and any other situation requiring synchronization of clocks.

SUMMARY OF THE INVENTION

Thus there is a particular need for a method and apparatus for generating controlled pulses on a periodic basis with fine resolution in the output number of pulses over long periods of time. There is also a need for a method and apparatus to provide improved clock synchronization at multiple locations.

A conventional technique for time synchronization, as described in Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Transactions on Communications,* vol. 39, no. 10, October 1991, pages 1482–93, is a "ping-pong" technique which uses round trip time tag messages to synchronize clocks which calculate the communications delays. A limitation of the ping-pong technique is that the difference between the delays in each direction between two terminals cannot be determined.

Commonly assigned Adamiak et al., U.S. application Ser. No. 08/713,295, "Digital Current Differential System," filed Sep. 13, 1996, describes a technique for compensating for this uncertainty in the embodiment of two or three terminal transmission lines by using information in the measured currents and digital communication. In this manner, measurement of magnitude and phase angle of power system voltages and currents at multiple locations can be performed

on a common time reference. When four or more terminals are used, the conventional ping-pong technique is used.

In one embodiment of the present invention, a method for synchronizing clocks at multiple terminals of a transmission line includes, at each local terminal: sensing a current; exchanging current and time stamp data between the local terminal and at least one remote terminal; estimating a frequency deviation between the frequency of a clock at the local terminal and a power system frequency using the sensed currents; estimating a time based phase deviation by using the time stamp data to compare a time delay between a transmission from the terminal to the at least one remote terminal with a time delay between a transmission from the at least one remote terminal to the terminal; estimating a current based phase deviation by determining a current phase angle deviation between the currents at the local terminal and the at least one remote terminal; and using the frequency deviation, the time based phase deviation, and the current based phase deviation to synchronize the clock at the local terminal.

A method for controlling a clock can include counting down by starting from a predetermined value and subtracting one (1) each time that a crystal produces an input pulse to provide a present count value; and repeating the following series of steps: producing an output pulse after the present count value reaches zero, determining a fraction representative of a difference between a desired output pulse rate and a time between beginning the counting down and producing the output pulse from the clock, adding the fraction to a fraction count value, if the fraction count value equals or exceeds one, decreasing the value of the fraction count value by one, and counting down by starting from a value equal to the predetermined value plus one and subtracting one each time that the crystal produces an input pulse to provide the present count value, and, if the fraction count value is less than one, counting down by starting from a value equal to the predetermined value and subtracting one each time that the crystal produces an input pulse to provide the present count value.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the invention believed to be novel are set forth with particularity in the appended claims. The invention itself, however, both as to organization and method of operation, together with further objects and advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings, where like numerals represent like components, in which:

FIG. 1 is a block diagram of a multi-terminal transmission line.

FIG. 2 is a block diagram of a clock synchronization embodiment for a two terminal system.

FIG. 3 is a block diagram of a loop filter useful in the embodiment of FIG. 2.

FIG. 4 is a block diagram of a clock and a counter of the present invention.

FIG. 5 is a graph of clock time with respect to real time according to one embodiment of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

FIG. 1 is a block diagram of a multi-terminal transmission line including two terminals **410** and **412** with a power line **414** and communication lines **416** therebetween. In a two

terminal system, two communication lines are useful in the event of a communication line failure. Current sensors **422** and **424** provide current signals to respective computers (shown as microprocessors **418** and **420**). In one embodiment, wherein the transmission lines have three phases, each of current sensors **422** and **424** includes three sensors with each sensor sensing a respective phase of the transmission line. The microprocessors can include clocks **10** and **110**.

FIG. **2** is a block diagram of a clock synchronization embodiment for a two terminal system. In one embodiment, distributed synchronization is accomplished by synchronizing the clocks to each other rather than to a master clock with the clocks being phase synchronized to each other and frequency synchronized to the power system frequency. Each microprocessor **418** and **420** compares the phase of its clock to the phase of the other clocks, compares the frequency of its clock to the power system frequency, and then makes appropriate adjustments. As long as there are enough operable communication lines, the clocks will be synchronized. Phase synchronization drives the relative timing error between clocks **10** and **110** to zero and is needed to control the uncertainty in the phase angle of phasor measurements. Frequency synchronization to the power system eliminates a source of error in phasor measurements that arises when data samples do not span exactly one cycle.

As shown in FIG. 2, each microprocessor **418** and **420** uses its respective frequency deviation calculator **212** or **214** to estimate the difference between its power system frequency and its clock frequency based on the rotation of phasors. For the present invention, it is not necessary to know the individual power system and clock frequency values—the difference between them is used. In a similar manner, phase deviation calculators **220**, **222**, **224**, and **226** provide phase deviation data when comparing the two terminals, but actual phase values are not needed to perform the deviation calculations.

Each microprocessor additionally uses its respective time based phase deviation calculator **220** or **222** and its respective current based phase deviation calculator **224** or **226** to estimate the time difference between its clock and the other clocks by exchanging information over communications channels. Phase and frequency loop filters **216** and **218** then use the frequency and phase angle deviation information to make fine adjustments to the clock frequencies.

As discussed in Adamiak et al., for conciseness, a phasor notation is used as follows:

$$\bar{I}(n)=\text{PhasorReal}_n+j\cdot\text{PhasorImaginary}_n, \quad (1)$$

$$\bar{I}_{a,k}(n)=\bar{I}(n) \text{ for phase a from the kth terminal at time step n}, \quad (2)$$

$$\bar{I}_{b,k}(n)=\bar{I}(n) \text{ for phase b from the kth terminal at time step n}, \quad (3)$$

$$\bar{I}_{c,k}(n)=\bar{I}(n) \text{ for phase c from the kth terminal at time step n}. \quad (4)$$

The positive sequence current can then be calculated for each terminal by the following equation:

$$\bar{I}_{pos,k}(n)=\tfrac{1}{3}\cdot\{\bar{I}_{a,k}(n)+e^{j2\pi/3}\bar{I}_{b,k}(n)+e^{-j2\pi/3}\bar{I}_{c,k}(n)\}, \quad (5)$$

wherein n is the sample number at the kth terminal of the transmission line.

As discussed in Adamiak et al., the contribution of charging currents can be removed at each respective terminal by subtraction. When a power system transmission line has line resistance and inductance, the sum of the currents entering a terminals is not exactly zero because of the capacitive charging current for the line. For short transmission lines, the charging current can be treated as an unknown error. In these embodiments, no voltage sensors are needed, and line charging current is included as a constant term in the total variance.

For long transmission lines, the charging current can become significant, so charging current compensation using voltage measurements is beneficial. One technique for such compensation is to subtract a C dv/dt term (capacitance multiplied by the change in voltage over time) from the measured current at each terminal of the system. This technique provides compensation of the capacitive current at both the fundamental power system frequency and some of the frequencies of the transient response of the transmission line. The fine details of traveling waves on the transmission line are not compensated for in this embodiment.

When three phase models are used, both phase to phase capacitance (Cpp) and phase to ground capacitance (Cpg) must be analyzed. In terms of zero sequence and positive sequence capacitance, Cpg and Cpp are given by Cpg=Czero (zero sequence capacitance) and Cpp=⅓ Cplus (positive sequence capacitance) minus ⅓ Czero. The compensation technique for each phase can use data from all three phases. For example, the compensation for phase "a" can be provided by Cpg*dVa/dt+Cpp*(2*dVa/dt−dVb/dt−dVc/dt), wherein Va, Vb, and Vc are phase voltages. Another equivalent expression for the phase "a" charging current is Cplus*(dVa/dt−dVo/dt)+Czero*dVo/dt, wherein Vo is the zero sequence voltage.

For some very long lines, the distributed nature of the lines leads to the classical transmission line equations, which can be solved for voltage and current profiles along the line. The compensation model uses the effective positive and zero sequence capacitance seen at the terminals of the line.

In some applications with long transmission lines, shunt reactors can be used to provide some of the charging current required by the line and to interact with the charging capacitance to introduce additional frequency components in the transient response of the transmission line. In one embodiment, the protection charging compensation is set to equal the residual charging current (the difference of capacitive and inductive reactance) at the fundamental power system frequency. The inductor current can be effectively "removed" from the circuit via a current transformer connection (not shown).

Time based phase deviation detection can be accomplished using the conventional ping-pong technique. The amount of time synchronization error in the ping-pong procedure depends on factors including the stability of the local clocks, how often the ping-pong is executed, and differential channel delay. The ping-pong must be executed often enough to compensate for drifts in the local clocks. A small amount of channel delay itself is not critical (and mainly affects only system transient response) provided that the channel delay is the same in each direction between terminals. If the channel delay is not the same, the difference between the delays causes a differential error between the clocks being synchronized over the restraint boundary and reduces the system sensitivity. If four or more terminals are used, the differential delay should be specified and controlled to achieve design goals.

Current based phase angle deviation detection can be accomplished for of two or three terminals by extracting additional information from the current phasors to determine phase angle errors. The basis for detecting deviations in clocks at the termination points of a transmission line is that,

according to fundamental circuit theory laws, the sum of the positive sequence currents is equal to the positive sequence charging current for the transmission lines. The positive sequence charging current can be calculated from measured voltages.

Inequalities are attributable to errors in the magnitudes and/or phase angles of the estimates of the positive sequence currents. In a two or a three terminal transmission line embodiment, phase angle errors, which depend on synchronization errors, can be determined approximately for each terminal.

As discussed above, for two or three terminal systems there are two separate sources of clock phase information: the exchange of time stamps over the communications channels and the current measurements themselves. Current measurements generally provide the most accurate information, but are not always available and may contain large errors during faults or switching transients as well as errors due to shunt capacitance current. Time stamped messages are the most reliable source of phase information but may suffer from a phase offset due to a difference in the channel delays in each direction between a pair of terminals. In some situations, one or both directions can be switched to a different physical path, leading to a phase error. For four or more terminals, the only source of phase information is the time tagged message exchange.

Time Based Phase Deviation

For time based phase deviation, during start up, the microprocessors can measure a minimum round trip delay, and during operation the phase error can be estimated as ½ of the absolute value of the difference between the round trip channel delay and the minimum round trip delay.

In one embodiment, the phase difference between a pair of clocks can be calculated by an exchange of time stamps as described in the aforementioned article by Mills. Each microprocessor exchanges time stamps with all other microprocessors that are accessible. It is not necessary to exchange time stamps with every terminal or to have all of the channels in operation. For each terminal that a given terminal can exchange time stamps, the clock deviation is calculated each time a complete set of time stamps arrives. A net deviation is the total deviation divided by the total number of terminals involved in the exchange. For example, in a two terminal embodiment, each respective microprocessor calculates a single time deviations from time stamps and divides the results by two. In a three terminal embodiment, each microprocessor calculates two time deviations and divides the result by three, If a channel is lost in a three terminal system, the single deviation that remains is divided by two.

As described by Mills, four time stamps are need to calculate round trip delay time and phase deviation. Three stamps are included in the message in each direction, and the fourth time stamp is the time when the message is received. Each time a message is received, the newest of the four time stamps are saved to become the first two time stamps of the next outgoing message, and the third time stamp of an outgoing message is the time when the message is transmitted. A fixed time shift is allowed between the stamp values and the actual events, provided the shift for outgoing message time stamps is the same for all terminals and the shift for incoming message time stamps is also identical.

For situations when the first message is transmitted by a given terminal or when an exchange is broken long enough to invalidate the last received set of time stamps (in one embodiment, about 66 milliseconds), the next outgoing set of time stamps may comprise a special start-up set containing transmittal time only. When such a message is received, nothing is calculated from it, except that the message time stamp and the time stamp when the message was received are saved for the next outgoing message.

The time stamp requirements are not very stringent because of the smoothing behavior of the phase locked loop. The time stamp can be a sample count with enough bits to cover the worst round trip including channel delay and processing delay. For example, an eight bit time stamp with one bit corresponding to a 1/64 of a power system cycle will accommodate a round trip delay of up to four cycles.

The round trip delay can be calculated by subtracting the delay between terminal 1 and terminal 2 (as measured by subtracting a first set of the four time stamps $T_{i-2}-T_{i-3}$) from the delay between terminal 2 and terminal 1 (as measured by subtracting a second set of the four time stamps $T_{i-1}-T_i$). The clock offset can be calculated by adding the two delays between the terminals and dividing by two. Although the round trip delay is a positive number, the clock offset can be positive or negative if time stamps are unsigned numbers that wrap around. If a roll over of any of the time stamps occurs, the calculations can be compensated. For example, if T_{i-2} is greater than T_{i-1}, then a predetermined number can be subtracted from the round trip delay and one half of this value can be subtracted from the clock offset, and if T_{i-3} is greater than T_i, then the predetermined number can be added to the round trip delay and one half of this value can be added to the clock offset. If these equations are calculated using integer values of time stamps, a conversion to phase angle in radians can be done by multiplying by 2π. The predetermined number is selected as the number of counts before which the clock rolls over. In one embodiment, for example, wherein there are 64 counts per a cycle and four cycles per a roll over, the predetermined number is 256.

Time stamp values can be obtained using software or hardware, provided that any jitter is limited to less than plus or minus about 130 microseconds. A fixed bias in the time stamp is acceptable if it is the same at each terminal.

Current Based Phase Deviation

As described in Adamiak et al., in a two terminal system, current phase angle deviations ($\phi_1(n)$, $\phi_2(n)$) can be calculated from the positive sequence currents as follows:

$$\phi_1(n) = \frac{1}{2} \cdot \arctan\left(\frac{\mathrm{imag}(\bar{I}_{pos,2}(n) \cdot \bar{I}^*_{pos,1}(n))}{\mathrm{real}(\bar{I}_{pos,2}(n) \cdot \bar{I}^*_{pos,1}(n))}\right), \text{ and} \tag{6}$$

$$\phi_2(n) = -\phi_1(n). \tag{7}$$

It is possible to use a four quadrant arc tangent, in which case the minus signs are needed on the imaginary and real part as shown.

In a three terminal system, the current phase angle deviations ($\phi_1(n)$, $\phi_2(n)$, $\phi_3(n)$) are approximated by the following equations:

$$\phi_1 \approx \frac{\mathrm{real}((\bar{I}_{pos,2}(n) - \bar{I}_{pos,3}(n)) \cdot (\bar{I}^*_{pos,1}(n) + \bar{I}^*_{pos,2}(n) + \bar{I}^*_{pos,3}(n)))}{\mathrm{imag}(\bar{I}_{pos,2}(n) \cdot \bar{I}^*_{pos,1}(n) + \bar{I}_{pos,3}(n) \cdot \bar{I}^*_{pos,2}(n) + \bar{I}_{pos,1}(n) \cdot \bar{I}^*_{pos,3}(n))}, \tag{8}$$

-continued

$$\phi_2 \approx \frac{\mathrm{real}((\bar{I}_{pos,3}(n) - \bar{I}_{pos,1}(n)) \cdot (\bar{I}^*_{pos,1}(n) + \bar{I}^*_{pos,2}(n) + \bar{I}^*_{pos,3}(n)))}{\mathrm{imag}(\bar{I}_{pos,2}(n) \cdot \bar{I}^*_{pos,1}(n) + \bar{I}_{pos,3}(n) \cdot \bar{I}^*_{pos,2}(n) + \bar{I}_{pos,1}(n) \cdot \bar{I}^*_{pos,3}(n))}, \tag{9}$$

$$\phi_3 \approx \frac{\mathrm{real}((\bar{I}_{pos,1}(n) - \bar{I}_{pos,2}(n)) \cdot (\bar{I}^*_{pos,1}(n) + \bar{I}^*_{pos,2}(n) + \bar{I}^*_{pos,3}(n)))}{\mathrm{imag}(\bar{I}_{pos,2}(n) \cdot \bar{I}^*_{pos,1}(n) + \bar{I}_{pos,3}(n) \cdot \bar{I}^*_{pos,2}(n) + \bar{I}_{pos,1}(n) \cdot \bar{I}^*_{pos,3}(n))}. \tag{10}$$

Frequency Deviation

Frequency deviations can be calculated by using the apparent rotation of phasors in the complex plane. The rotational rate of phasors is equal to the difference between the power system frequency and the ratio of the sampling frequency divided by the number of samples per cycle.

As described in Adamiak et al., to determine frequency deviations, for each terminal a quantity can be derived from the positive sequence current (with or without removal of the charging current—depending on the application) that is indicative of the amount of rotation from one cycle to the next by calculating the product of the positive sequence current and the complex conjugate of the positive sequence current from the previous cycle:

$$\mathrm{Deviation} = \bar{I}_{pos,k}(n) \cdot ((\bar{I}_{pos,k}(n-N))^* \tag{11}$$

The angle of the deviation phasor per cycle for each terminal is proportional to the frequency deviation at that terminal, as discussed in commonly assigned Premerlani, U.S. Pat. No. 4,715,000, issued Dec. 22, 1987.

The frequency deviation is calculated from the deviation phasor Deviation:

$$\frac{\Delta f}{f_o} = \arctan\left(\frac{\frac{\mathrm{imag(Deviation)}}{\mathrm{real(Deviation)}}}{2\pi}\right) \tag{13}$$

wherein Δf is the frequency deviation and f_o is the nominal (system) frequency. A four quadrant arc tangent can be calculated by taking the imaginary and the real part of the deviation separately for the two arguments of the four quadrant arc tangent. Preferably a radian frequency is used in the respective loop filter. The radian frequency can then obtained by the following equation:

$$\Delta\omega = \Delta f \cdot 2\pi. \tag{14}$$

Or, equations (13) and (14) can be rewritten as follows:

$$\Delta\omega = f_o \cdot \arctan\left(\frac{\mathrm{imag(Deviation)}}{\mathrm{real(Deviation)}}\right). \tag{15}$$

The frequency deviation information from frequency deviation calculators **212** and **214**, the time based phase deviation information from time based phase deviation calculators **220** and **222**, and, for two or three terminal systems, the current based phase deviation information from current based phase deviation calculators **224** and **226** is supplied to respective loop filters **216** and **218**.

FIG. **3** is a block diagram of a loop filter **216** useful in the embodiment of FIG. **2** wherein a phase information selector

410 determines which of the phase deviation data (time based phase deviation information delta phi time or current based phase deviation information delta phi current) to use. Generally, if available, the current based phase deviation is more precise and is the information which is selected. If the phase deviation is too large, however, the current based phase deviation information may not accurately reflect the large deviation. Therefore, in a preferred embodiment, the phase information selector determines whether the time based phase deviation is greater than a predetermined threshold. Then, even if current based phase deviation information is available, the phase information selector selects the time based phase deviation. In one embodiment, the predetermined time period is one half a cycle.

The primary feedback mechanism shown in FIG. **3** is the use of the selected time or current based deviation information by a proportional plus integral (PI) filter (shown in FIG. **3** as a combination of an integrator **232** and a gain element **236**). The PI filter further includes a frequency deviation input signal (delta omega) from an gain element **228** and adder **230** to provide frequency tracking.

Depending on the gains of the PI filter's proportional term and integral term, the transient behavior of the loop can be under damped, critically damped, or over damped. In the present invention, it is preferred that the transient behavior is critically damped. The loop time constant should be selected by considering the goal of critical damping and the effects of both phase and frequency noise. In one embodiment, the time constant for the PI filter main loop is about 10 seconds.

A secondary loop is formed through the frequency deviation input signal of the filter. If a frequency deviation input signal is available, the signal is integrated because frequency is the derivative of phase information. Preferably, the integrator of the frequency deviation input signal is the same integrator **232** of the selected time or current based phase deviation input signal. In one embodiment, the frequency deviation input signal passes through gain element **228** before being added with adder **230** to the signal to be integrated by integrator **232**. It is useful to have a single combined integrator because if two separate integrators are used, they can drift in opposite directions into saturation with the loop driving their sum to zero.

In normal operation, frequency tracking at each terminal matches the tracking at all other terminals because all terminals will measure approximately the same frequency deviation. If there is not enough current at a terminal to estimate frequency deviation, frequency tracking at that terminal can be accomplished indirectly via phase locking to other terminals. A small phase deviation must be present for the tracking to occur. To keep the deviation from exceeding a target of about 0.01 radians, the slew rate of frequency tracking should be limited to about 0.0001 hertz per second. With a worst case step change of 0.1 hertz, the time constant of frequency tracking should be at least about 1000 seconds.

Also shown in loop filter **216** is clock **10**. The clock behaves in a similar manner as an integrator. In the embodiment of FIG. **3**, the clock receives a signal from adder **240** which represents the combination of the output signal of PI filter **232** and the selected time or current based phase deviation signal after passing through gain element **238**. The signal from adder **240** is used to adjust the frequency of the clock. The clock can be implemented in hardware and software with a crystal oscillator and a counter as discussed with respect to FIGS. **4** and **5**, for example.

If the ratio of the time step of the integrators (1/60 second, for example) to the shortest time constant (10 seconds, for

example) is small (1/600, for example), the integrators can be implemented simply as the simple summations with a gain multiplier of the time step (1/60 second, for example).

There are three gains that must be selected for system design. These gains are determined by the time step of the integrators and the desired time constants of the system as follows:

$$KI = \frac{T_{repeat}}{T_{phase}{}^2}$$

$$KP = \frac{2}{T_{phase}}$$

$$KF = \frac{T_{repeat}}{T_{frequency}}.$$

T_{repeat}=time between executions of the filter algorithm
T_{phase}=time constant for primary phase locked loop
$T_{frequency}$=time constant for frequency locked loop
In one embodiment, the time step for the integrators is 1/60 of a second, and the time constants are 10 seconds for the time stamp phase locking and 1000 seconds for the frequency tracking.

FIG. **4** is a block diagram of a clock **10** and a counter **12** of the present invention, and FIG. **5** is an graph of clock time with respect to real time according to one embodiment of the present invention. The counter can be separate from or included in a computer **14**. The computer uses a desired output pulse rate to calculate a counter value that includes both an integer portion and a fractional portion. In one embodiment, the counter includes a hardware counter **16** for storing the integer portion and a software counter 18 for storing the fractional portion.

As discussed above, the resolution in the output rate of conventional counters is limited by the pulse frequency of the crystal. Thus a digital clock will generally be operating either too fast or too slow with respect to real time. In the example of FIG. **5**, $rate_x$ has a slope which is too low while $rate_{x+1}$ has a slope which is too high. The present invention switches between $rate_x$ and $rate_{x+1}$ in a manner designed to bring the clock closer to the target rate than to $rate_x$ or $rate_{x+1}$.

The resolution of the fractional portion of the clock can be chosen to provide the desired resolution for the time period of the output pulses. The integer portion of the output pulse rate is used as the load value of the hardware counter **16** at the beginning of each output time period.

The fractional portion of the output pulse rate is accumulated into the software counter **18** each time the integer portion is reloaded. The software counter acts as a fractional accumulator with wrap around properties. For example, a value of 0.6 added to 0.8 would produce a 0.4, with an overflow. Each time an overflow is produced, the counter value for the next time period is increased by one count over that indicated by the integer portion of the counter reload value for that period only. The fractional register is not reset, even when the desired period changes.

Thus, the lower rate ($rate_x$) is first used. After an overflow is produced (A), the counter value is increased by 1 and $rate_{x+1}$ is used for the next time period. If an overflow is again produced in the next time period (B), then the counter value is then again increased by 1 and $rate_{x+1}$ is again used for the following time period. If an overflow is then not produced (C), the counter value is not increased, and the lower $rate_x$ is used until an overflow is produced (D). The present invention will switch between $rate_x$ and $rate_{x+1}$ in a manner so that the integral of the rate of accumulated error between the clock rate and the target rate is driven towards zero.

The net effect of the invention is that the time between output pulses takes on two values, one that is slightly lower than the desired value, and one that is slightly higher, in a slowly repeating pattern with an accumulated error that approaches zero over a long time period.

In one embodiment, a 6 megahertz crystal clock and a 16 bit counter are used. The counter is loaded with a desired period, which is four data samples in one embodiment. Each time the period is counted out, data is sampled. After four samples (1/16 of a cycle), the counter is reloaded. Time periods between data samples are calculated as a 32 bit multiple of the period of the 6 megahertz clock with a 16 bit integer and a 16 bit fraction. Two separate 16 bit registers are used to control the clock. One register controls the integer portion of the time period, and the other controls the fractional portion. The integer register is used to reload the hardware counter every four samples.

While only certain preferred features of the invention have been illustrated and described herein, many modifications and changes will occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the invention.

I claim:

1. A method for synchronizing clocks at multiple terminals of a transmission line, the method comprising, at each local terminal:
 - sensing a current at the local terminal;
 - exchanging current and time stamp data between the local terminal and at least one remote terminal;
 - estimating a frequency deviation between the frequency of a clock at the local terminal and a power system frequency using the sensed currents;
 - estimating a time based phase deviation by using the time stamp data to compare a time delay between a transmission from the local terminal to the at least one remote terminal with a time delay between a transmission from the at least one remote terminal to the local terminal;
 - estimating a current based phase deviation by determining a current phase angle deviation between the currents at the local terminal and the at least one remote terminal;
 - using the frequency deviation, the time based phase deviation, and the current based phase deviation to synchronize the clock at the local terminal.

2. The method of claim 1, wherein sensing a current at the local terminal comprises sensing three phase currents at the local terminal and calculating a positive sequence current using the three phase currents.

3. The method of claim 2, wherein using the frequency deviation, the time based phase deviation, and the current based phase deviation to synchronize the clock at the local terminal includes:
 - selecting a representative phase deviation from the time based phase deviation and the current based phase deviation;
 - adding a first phase multiple of the representative phase deviation and a frequency multiple of the frequency deviation to provide a summed deviation;
 - integrating the summed deviation;
 - adding the integrated summed deviation and a second phase multiple of the representative phase deviation to provide a frequency signal; and
 - sending the frequency signal to the clock.

4. The method of claim 3, wherein selecting the representative phase deviation from the time based phase deviation and the current based phase deviation comprises:

determining whether the time based phase deviation exceeds a predetermined threshold;

determining the availability of current based phase deviation;

if the current based phase deviation is not available or if the time based phase deviation does exceed the predetermined threshold, selecting the time based phase deviation as the representative phase deviation; and

if the current based phase deviation is available and if the time based phase deviation does not exceed the predetermined threshold, selecting the current based phase deviation as the representative phase deviation.

5. An apparatus for synchronizing clocks at multiple terminals of a transmission line, the apparatus comprising, at each local terminal:

a current sensor for sensing a current at the local terminal;

a communication path for exchanging current and time stamp data between the local terminal and at least one remote terminal; and

a computer for:

estimating a frequency deviation between the frequency of a clock at the local terminal and a power system frequency using the sensed currents,

estimating a time based phase deviation by using the time stamp data to compare a time delay between a transmission from the local terminal to the at least one remote terminal with a time delay between a transmission from the at least one remote terminal to the local terminal,

estimating a current based phase deviation by determining a current phase angle deviation between the currents at the local terminal and the at least one remote terminal, and

using the frequency deviation, the time based phase deviation, and the current based phase deviation to synchronize the clock at the local terminal.

6. The apparatus of claim 5, wherein the current sensor comprises three phase current sensors at the local terminal and the computer is adapted to calculate a positive sequence current using three phase currents sensed by the three phase current sensors.

7. The apparatus of claim 6, wherein the computer is adapted to use the frequency deviation, the time based phase deviation, and the current based phase deviation to synchronize the clock at the local terminal by:

selecting a representative phase deviation from the time based phase deviation and the current based phase deviation;

adding a first phase multiple of the representative phase deviation and a frequency multiple of the frequency deviation to provide a summed deviation;

integrating the summed deviation;

adding the integrated summed deviation and a second phase multiple of the representative phase deviation to provide a frequency signal; and

sending the frequency signal to the clock.

8. The apparatus of claim 7, wherein the computer is adapted to select the representative phase deviation from the time based phase deviation and the current based phase deviation by:

determining whether the time based phase deviation exceeds a predetermined threshold;

determining the availability of current based phase deviation;

if the current based phase deviation is not available or if the time based phase deviation does exceed the predetermined threshold, selecting the time based phase deviation as the representative phase deviation; and

if the current based phase deviation is available and if the time based phase deviation does not exceed the predetermined threshold, selecting the current based phase deviation as the representative phase deviation.

9. A method for using a crystal clock to achieve a desired output pulse rate, the method comprising:

determining two count rates which are respectively higher than and lower than the desired output pulse rate;

producing an output pulse after counting through a first one of the two count rates; and repeating the following series of steps:

determining a fraction representative of a difference between the desired output pulse rate and a time period elapsed while counting,

adding the fraction to a fraction count value,

if a magnitude of the fraction count value is greater than or equal to one, removing an integer portion of the fraction count value and producing the output pulse after counting through a second one of the two count rates, and

if a magnitude of the fraction count value is less than one, producing the output pulse after counting through the first one of the two count rates.

10. An apparatus for controlling a crystal clock, the apparatus comprising:

a computer for using two count rates which are respectively higher than and lower than the desired output pulse rate, and producing an output pulse after counting through a first one of the two count rates,

the computer adapted to repeat the following series of steps:

determining a fraction representative of a difference between the desired output pulse rate and a time period elapsed while counting,

adding the fraction to a fraction count value,

if a magnitude of the fraction count value is greater than or equal to one, removing an integer portion of the fraction count value and producing the output pulse after counting through a second one of the two count rates, and

if a magnitude of the fraction count value is less than one, producing the output pulse after counting through the first one of the two count rates.

* * * * *

US005757772A

United States Patent [19]

Thornberg et al.

[11] **Patent Number: 5,757,772**

[45] **Date of Patent: May 26, 1998**

[54] **PACKET SWITCHED RADIO CHANNEL TRAFFIC SUPERVISION**

[75] Inventors: **Carl Magnus Thornberg; Magnus Andersson**, both of Stockholm; **Olle Erik Grimlund**, Bromma, all of Sweden

[73] Assignee: **Telefonaktiebolaget LM Ericsson**, Stockholm, Sweden

[21] Appl. No.: **581,475**

[22] Filed: **Oct. 24, 1995**

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 529,559, Sep. 18, 1995.

[51] **Int. Cl.**[6] **H04J 3/14**

[52] **U.S. Cl.** **370/236**; 370/252

[58] **Field of Search** 370/17, 60, 60.1, 370/85.6, 85.7, 94.1, 95.1, 229, 230, 231, 232, 234, 235, 236, 252, 253, 328, 329, 332, 426; 455/33.1, 34.1, 34.2, 53.1, 54.1, 422, 423, 434, 517; 379/59, 63

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,984,264	1/1991	Katsube	379/197
5,050,161	9/1991	Golestani	370/60
5,090,011	2/1992	Fukuta et al.	370/60
5,131,020	7/1992	Liebesny et al.	379/59
5,220,674	6/1993	Morgan et al.	395/800
5,276,677	1/1994	Ramamurthy et al.	370/60
5,335,222	8/1994	Kamoi et al.	370/60
5,357,507	10/1994	Hughes et al.	370/60
5,400,329	3/1995	Tokura et al.	370/84
5,497,504	3/1996	Acampora et al.	455/33.2
5,521,925	5/1996	Merakos et al.	370/337
5,533,024	7/1996	Owada	370/84
5,537,395	7/1996	Alles et al.	370/17

FOREIGN PATENT DOCUMENTS

0 332 818	1/1989	European Pat. Off. .
0 415 898	8/1990	European Pat. Off. .
WO 95/16330	6/1995	WIPO .

OTHER PUBLICATIONS

XV International Switching Symposium, vol. 1, 23–28 Apr. 1995, Berlin, DE, pp. 246–250, XP000495573 Bianchi, et al.: "Dynamic Channel Allocation Procedures for Packet Data Services over GSM Networks".

Primary Examiner—Alpus H. Hsu
Assistant Examiner—Kwang Bin Yao
Attorney, Agent, or Firm—Jenkens & Gilchrist, P.C.

[57] **ABSTRACT**

A method and system for packet switched radio channel (PRCH) traffic supervision is disclosed. A PRCH supervision function receives a packet report for each data packet transmitted on the PRCH. The PRCH supervision function calculates an estimate of average data traffic for each packet call on the PRCH, an estimate of average data traffic on the PRCH and an estimate of the average packet delay on the PRCH. The calculations may be done for the uplink and downlink of the PRCH separately, or, as values for the combined uplink and downlink of the PRCH. The results of the calculations may then be used to determine if a packet call should be admitted to the PRCH or if a packet call should be expelled from the PRCH.

19 Claims, 15 Drawing Sheets

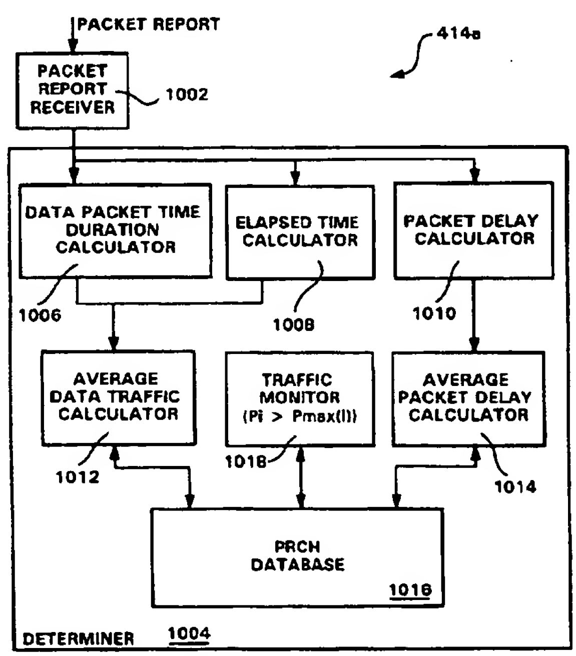


FIG. 1

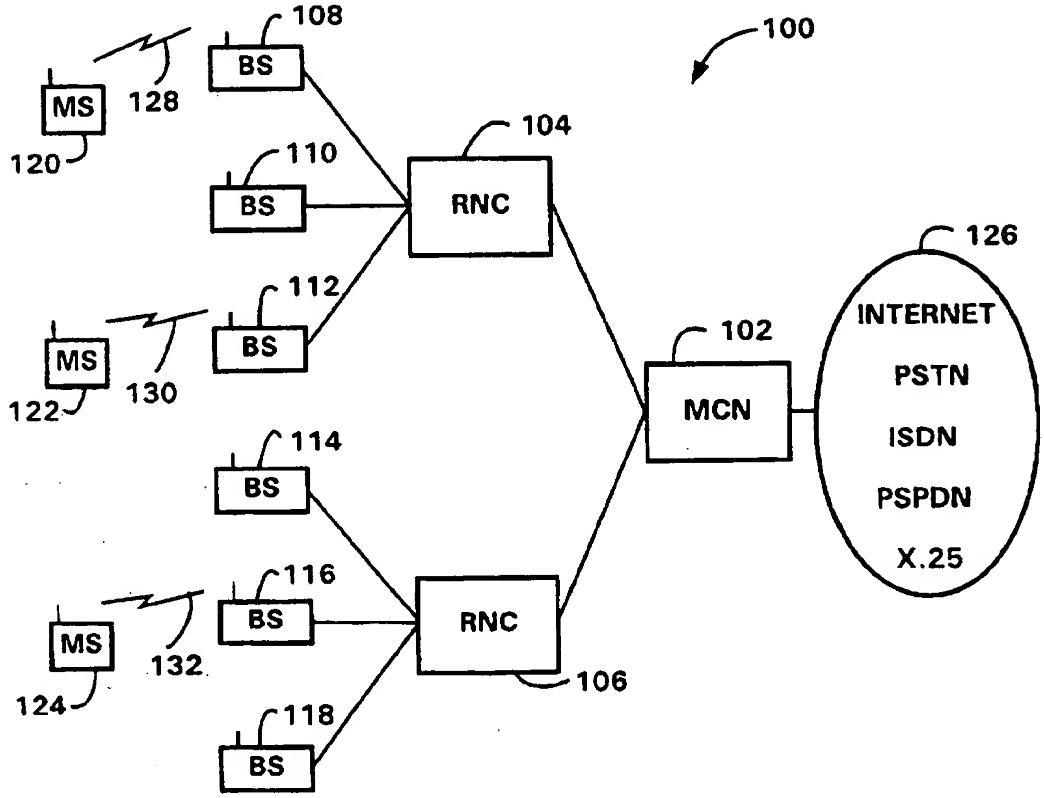

100
108
BS
128
MS
120
104
110
BS
RNC
126
112
INTERNET
MS
BS
102
PSTN
122
130
MCN
ISDN
114
BS
PSPDN
116
X.25
MS
BS
RNC
124
132
106
118
BS

FIG. 2

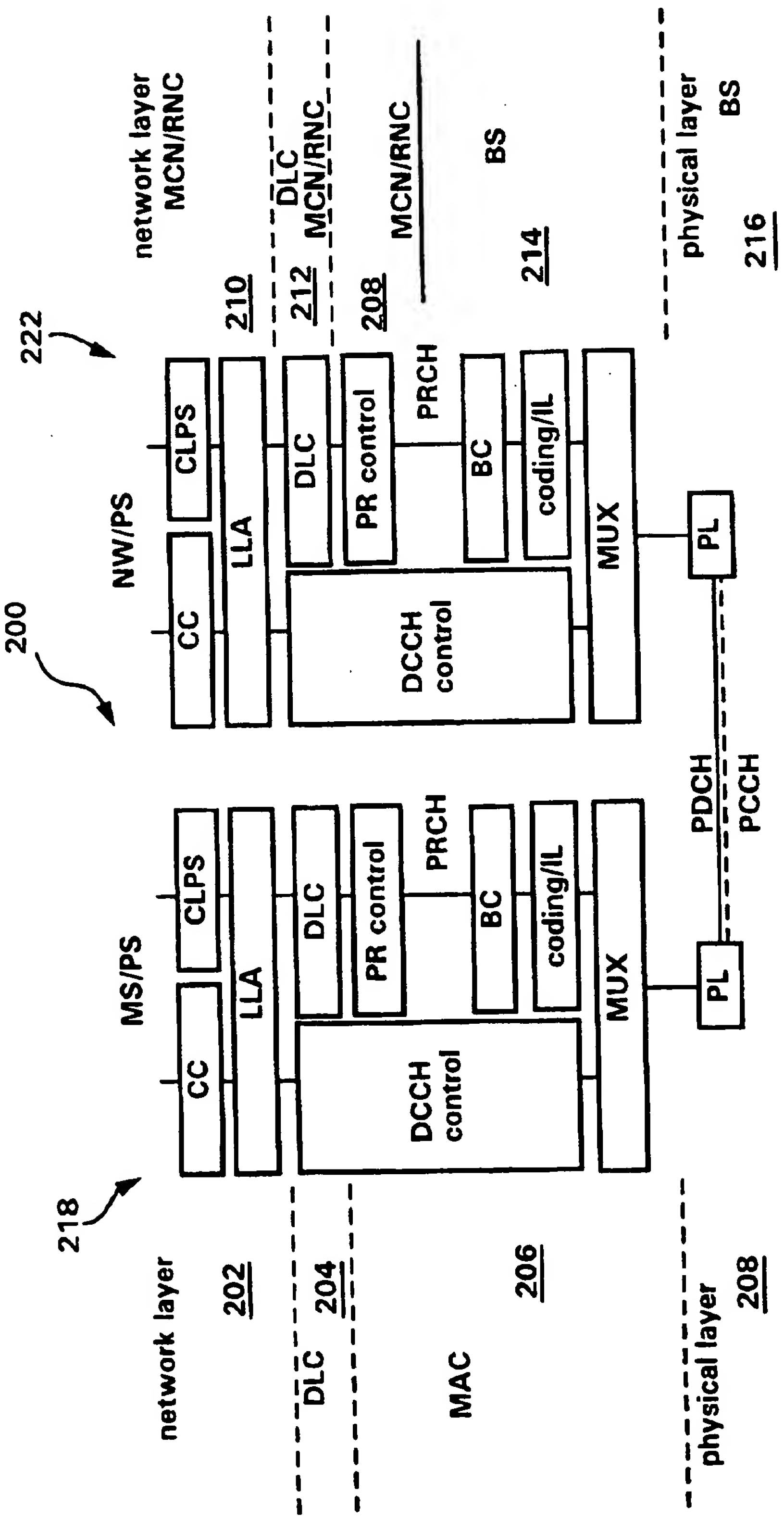

200
222
218
NW/PS
MS/PS
CC
CLPS
LLA
DLC
PR control
PRCH
BC
coding/IL
DCCH control
MUX
PL
PDCH
PCCH
network layer MCN/RNC
210
DLC MCN/RNC
212
208
MCN/RNC
BS
214
physical layer BS
216
network layer
202
DLC
204
MAC
206
physical layer
208

FIG. 3A

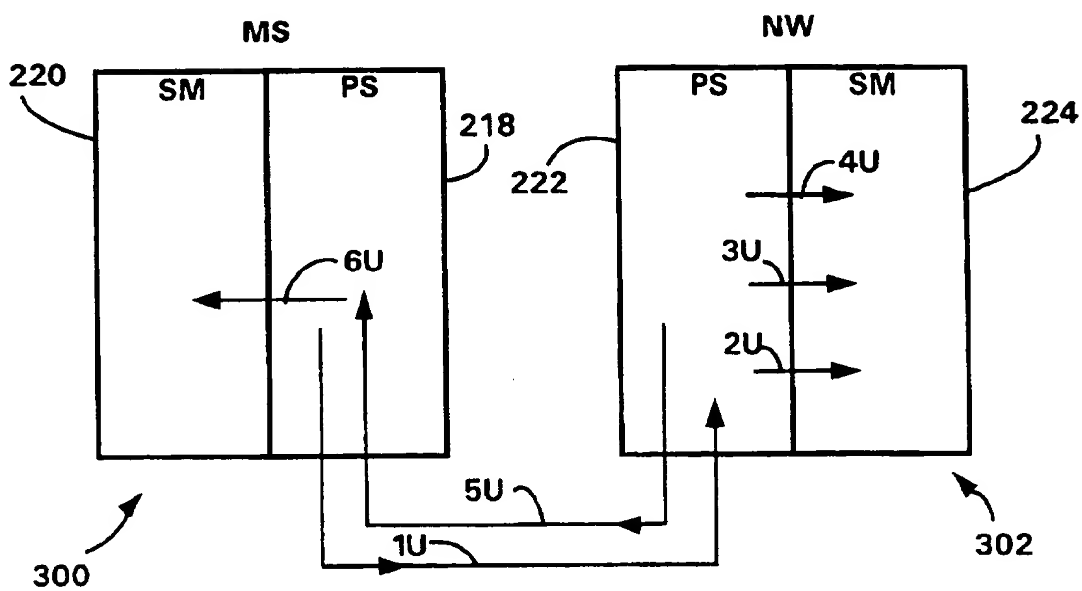

MS
NW
220
SM
PS
218
PS
SM
224
222
4U
6U
3U
2U
5U
1U
300
302

FIG. 3B

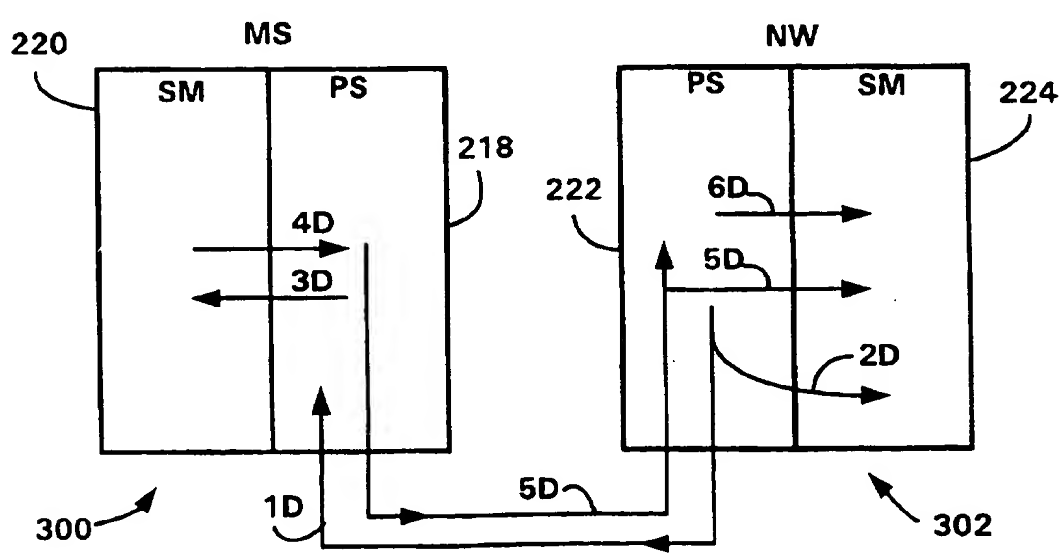

220
MS
NW
SM
PS
PS
SM
224
218
222
6D
4D
5D
3D
2D
5D
1D
300
302

FIG. 4

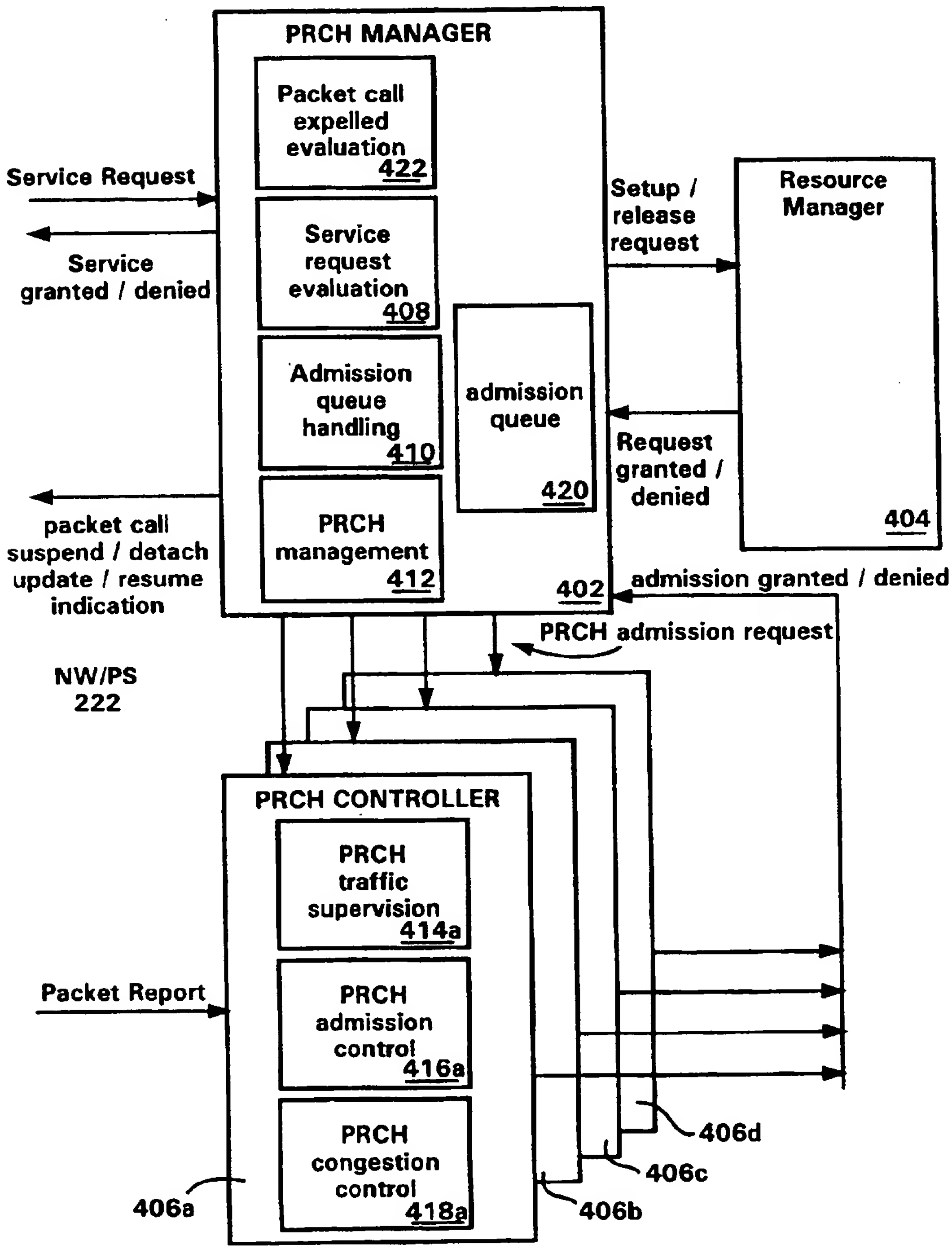

PRCH MANAGER
Packet call expelled evaluation 422
Service request evaluation 408
Admission queue handling 410
PRCH management 412
admission queue 420
402
Service Request
Service granted / denied
packet call suspend / detach update / resume indication
NW/PS 222
Setup / release request
Resource Manager 404
Request granted / denied
admission granted / denied
PRCH admission request
PRCH CONTROLLER
PRCH traffic supervision 414a
PRCH admission control 416a
PRCH congestion control 418a
Packet Report
406a
406b
406c
406d

FIG.5A

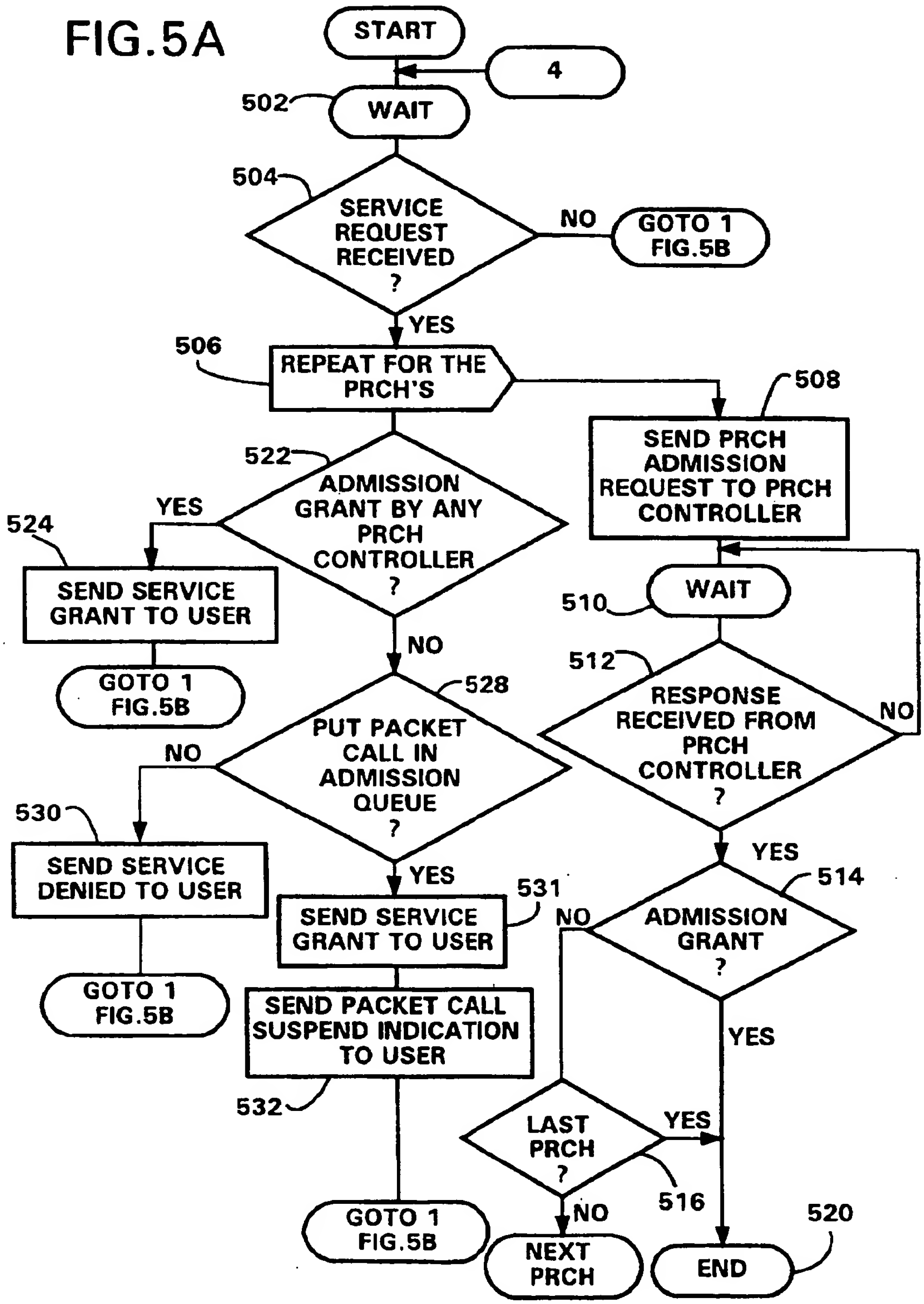
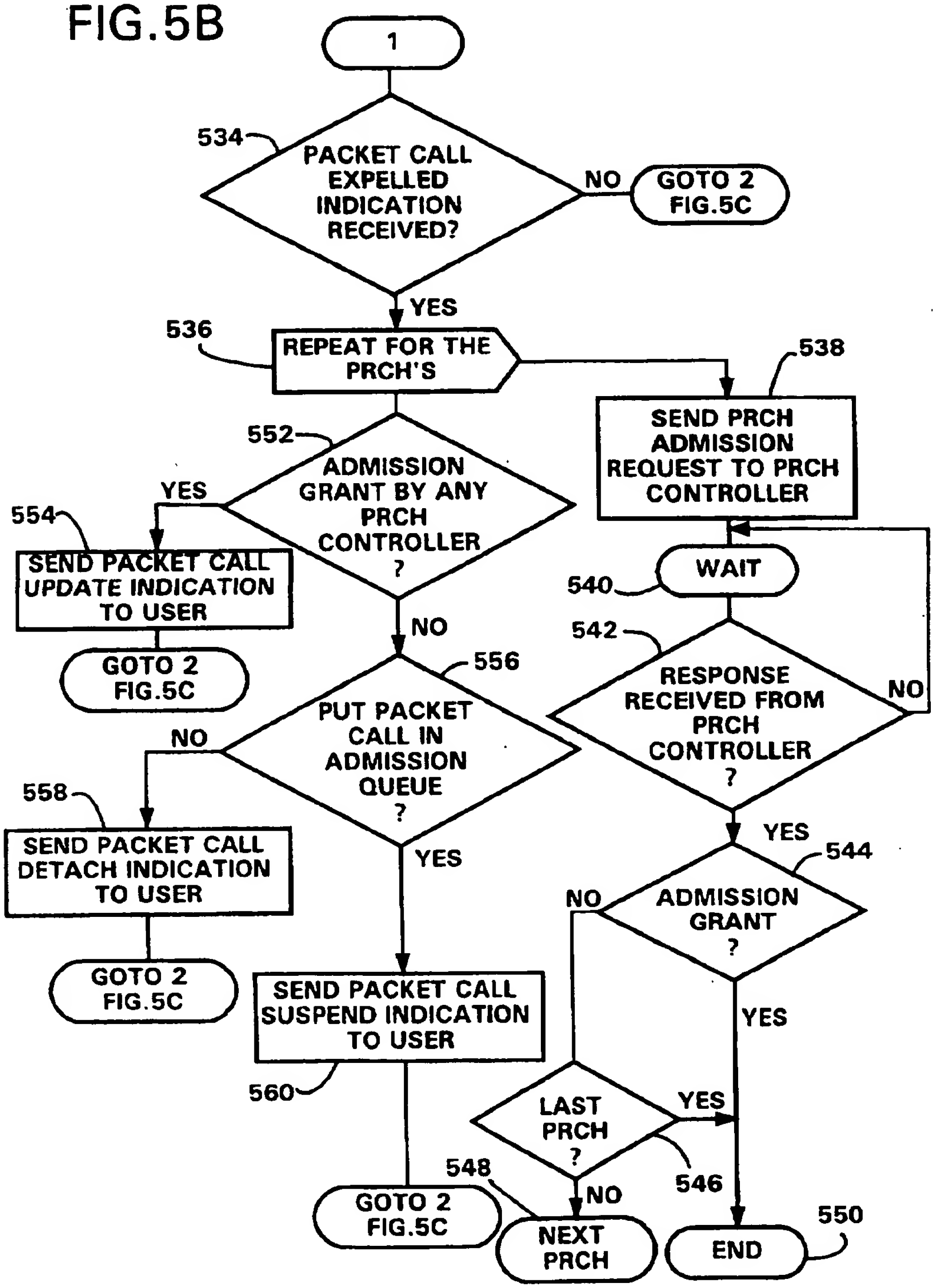

START
4
502
WAIT
504
SERVICE REQUEST RECEIVED ?
NO
GOTO 1 FIG.5B
YES
506
REPEAT FOR THE PRCH'S
508
SEND PRCH ADMISSION REQUEST TO PRCH CONTROLLER
522
ADMISSION GRANT BY ANY PRCH CONTROLLER ?
YES
524
SEND SERVICE GRANT TO USER
GOTO 1 FIG.5B
NO
510
WAIT
512
RESPONSE RECEIVED FROM PRCH CONTROLLER ?
NO
528
PUT PACKET CALL IN ADMISSION QUEUE ?
NO
530
SEND SERVICE DENIED TO USER
GOTO 1 FIG.5B
YES
SEND SERVICE GRANT TO USER
531
YES
514
ADMISSION GRANT ?
NO
SEND PACKET CALL SUSPEND INDICATION TO USER
532
YES
LAST PRCH ?
YES
516
NO
GOTO 1 FIG.5B
NEXT PRCH
END
520

FIG.5B



1
534
PACKET CALL EXPELLED INDICATION RECEIVED?
NO
GOTO 2 FIG.5C
YES
536
REPEAT FOR THE PRCH'S
538
SEND PRCH ADMISSION REQUEST TO PRCH CONTROLLER
552
ADMISSION GRANT BY ANY PRCH CONTROLLER ?
YES
554
SEND PACKET CALL UPDATE INDICATION TO USER
GOTO 2 FIG.5C
540
WAIT
542
RESPONSE RECEIVED FROM PRCH CONTROLLER ?
NO
NO
556
PUT PACKET CALL IN ADMISSION QUEUE ?
NO
558
SEND PACKET CALL DETACH INDICATION TO USER
GOTO 2 FIG.5C
YES
YES
544
ADMISSION GRANT ?
NO
YES
SEND PACKET CALL SUSPEND INDICATION TO USER
560
LAST PRCH ?
YES
546
548
NO
GOTO 2 FIG.5C
NEXT PRCH
END
550

FIG.5C

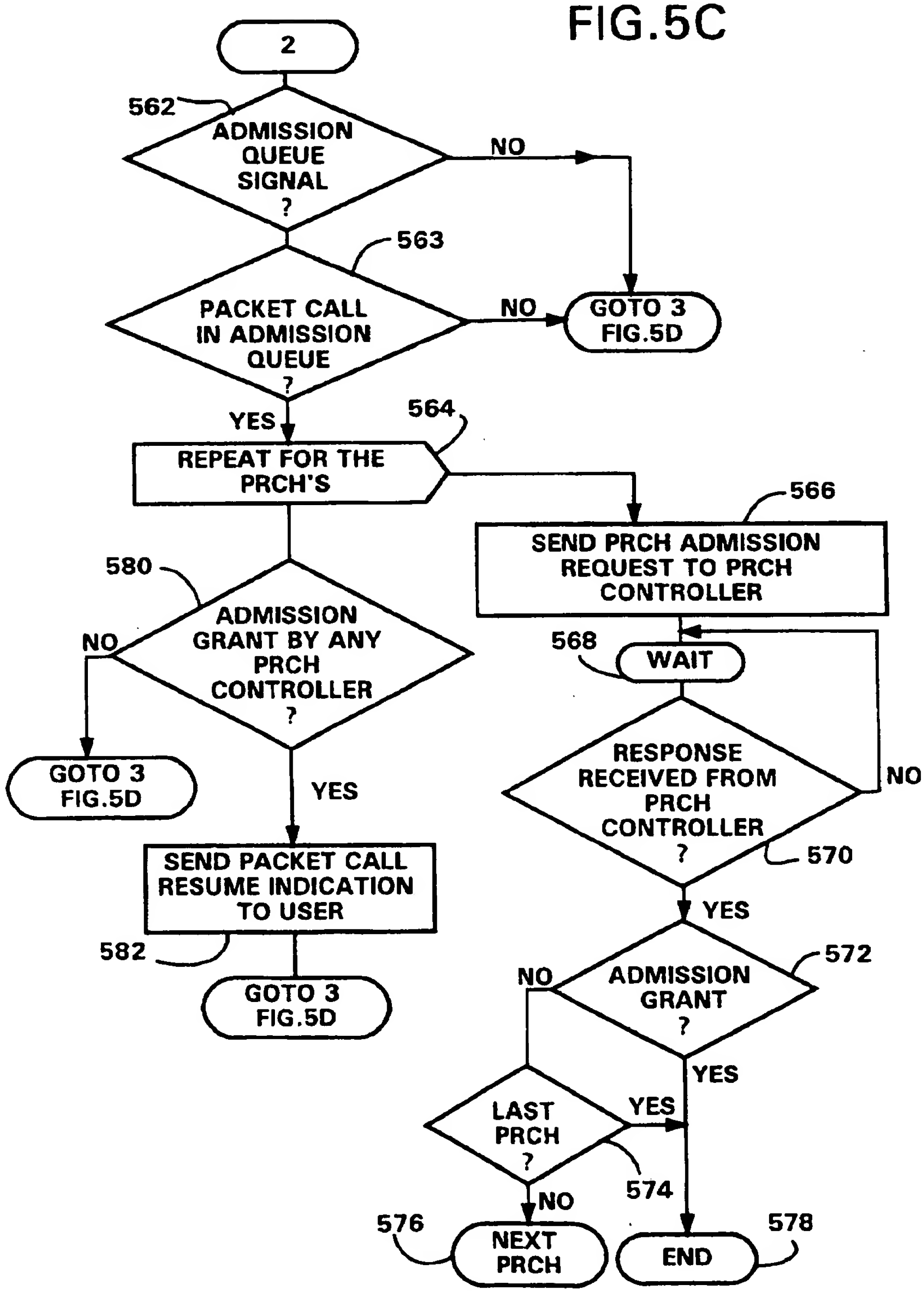

2
562
ADMISSION QUEUE SIGNAL ?
NO
563
PACKET CALL IN ADMISSION QUEUE ?
NO
GOTO 3 FIG.5D
YES
564
REPEAT FOR THE PRCH'S
566
SEND PRCH ADMISSION REQUEST TO PRCH CONTROLLER
580
ADMISSION GRANT BY ANY PRCH CONTROLLER ?
NO
568
WAIT
GOTO 3 FIG.5D
YES
RESPONSE RECEIVED FROM PRCH CONTROLLER ?
NO
570
SEND PACKET CALL RESUME INDICATION TO USER
YES
582
572
NO
ADMISSION GRANT ?
GOTO 3 FIG.5D
YES
LAST PRCH ?
YES
574
NO
576
NEXT PRCH
END
578

FIG.5D

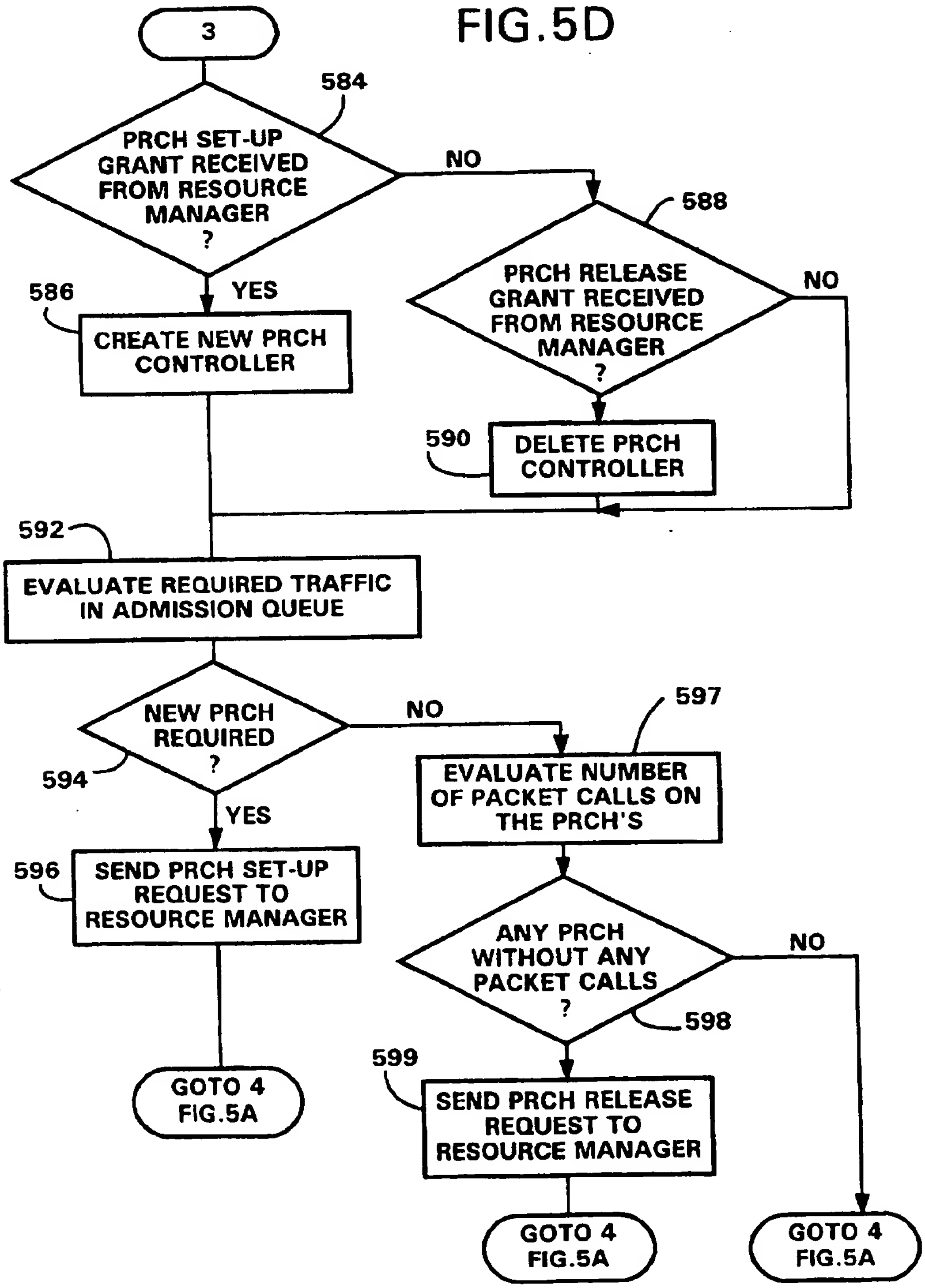
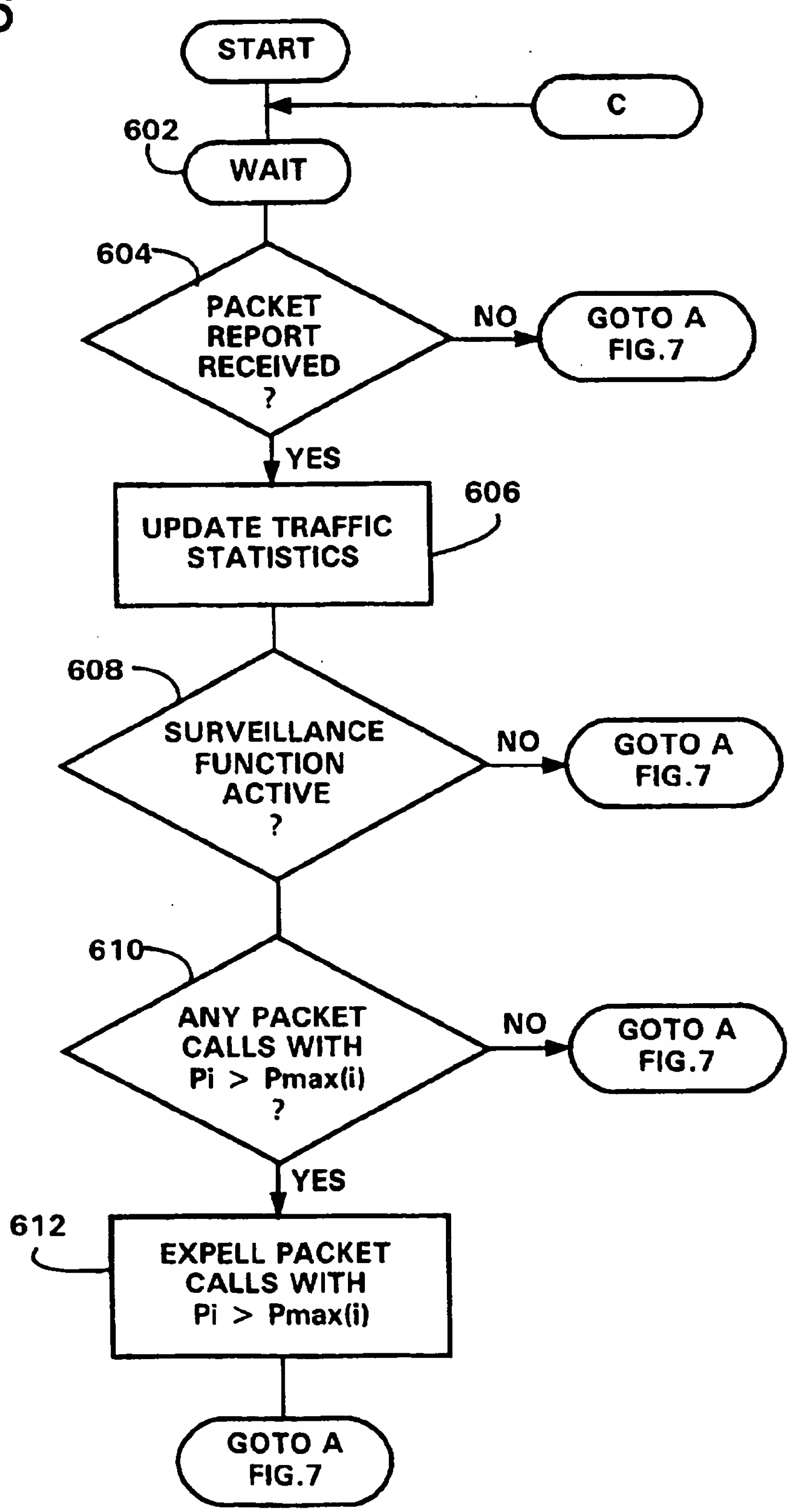

3
584
PRCH SET-UP GRANT RECEIVED FROM RESOURCE MANAGER ?
NO
588
PRCH RELEASE GRANT RECEIVED FROM RESOURCE MANAGER ?
NO
YES
586
CREATE NEW PRCH CONTROLLER
590
DELETE PRCH CONTROLLER
592
EVALUATE REQUIRED TRAFFIC IN ADMISSION QUEUE
NEW PRCH REQUIRED ?
NO
597
594
YES
EVALUATE NUMBER OF PACKET CALLS ON THE PRCH'S
596
SEND PRCH SET-UP REQUEST TO RESOURCE MANAGER
ANY PRCH WITHOUT ANY PACKET CALLS ?
NO
598
599
GOTO 4 FIG.5A
SEND PRCH RELEASE REQUEST TO RESOURCE MANAGER
GOTO 4 FIG.5A
GOTO 4 FIG.5A

FIG.6



START
C
602
WAIT
604
PACKET REPORT RECEIVED ?
NO
GOTO A FIG.7
YES
606
UPDATE TRAFFIC STATISTICS
608
SURVEILLANCE FUNCTION ACTIVE ?
NO
GOTO A FIG.7
610
ANY PACKET CALLS WITH Pi > Pmax(i) ?
NO
GOTO A FIG.7
YES
612
EXPELL PACKET CALLS WITH Pi > Pmax(i)
GOTO A FIG.7

FIG.7

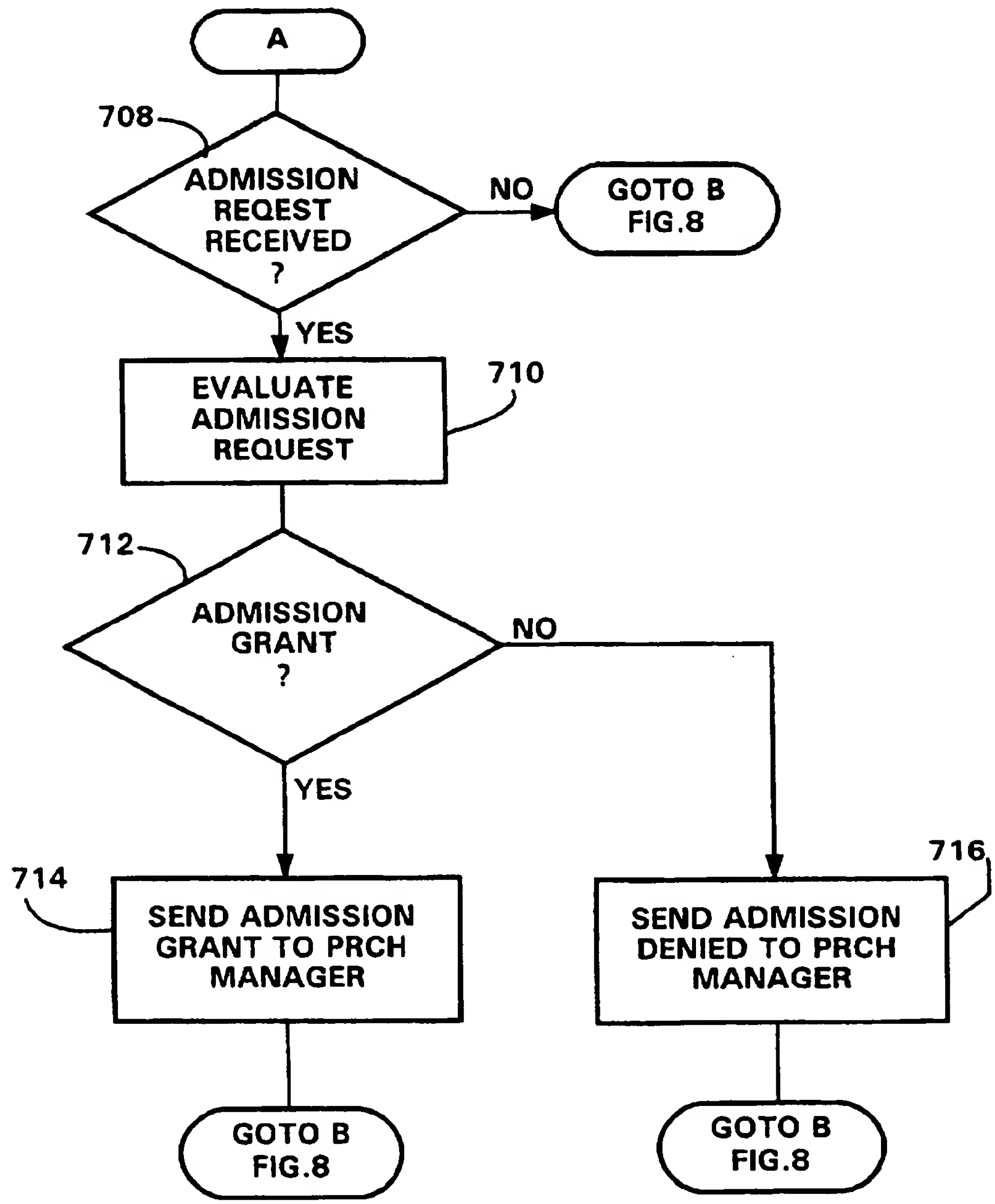
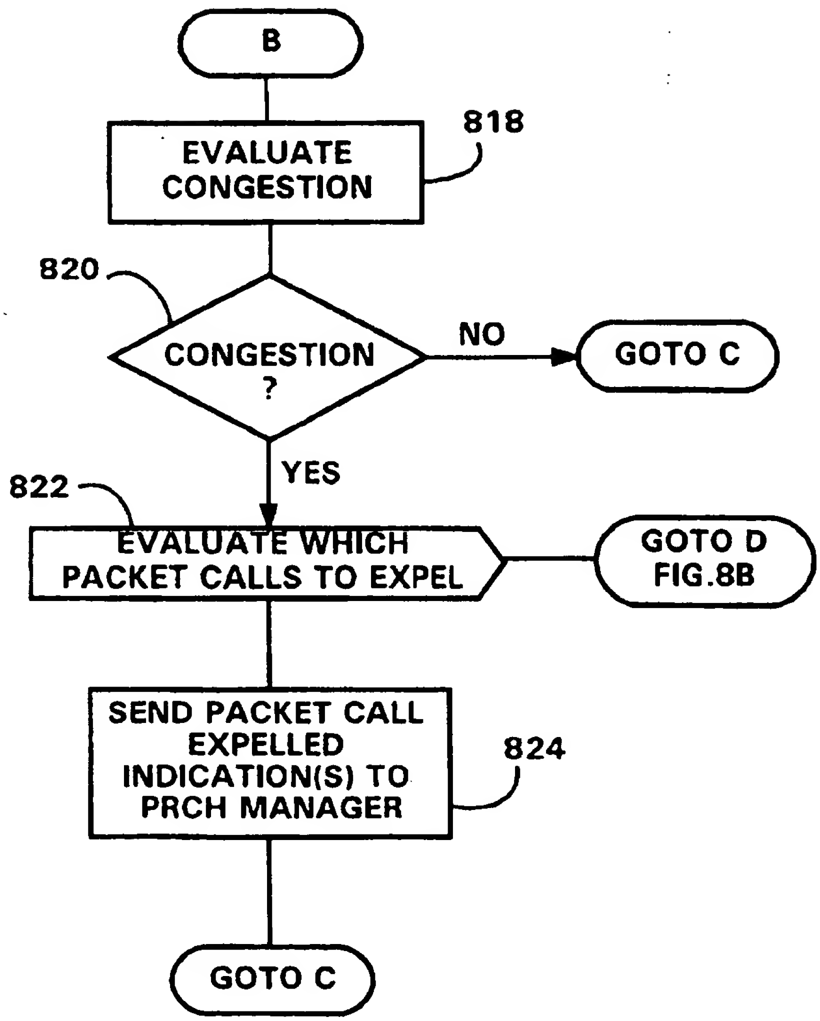

A
708
ADMISSION REQEST RECEIVED ?
NO
GOTO B FIG.8
YES
EVALUATE ADMISSION REQUEST
710
712
ADMISSION GRANT ?
NO
YES
714
SEND ADMISSION GRANT TO PRCH MANAGER
716
SEND ADMISSION DENIED TO PRCH MANAGER
GOTO B FIG.8
GOTO B FIG.8

FIG.8A



B
EVALUATE CONGESTION
818
820
CONGESTION ?
NO
GOTO C
YES
822
EVALUATE WHICH PACKET CALLS TO EXPEL
GOTO D FIG.8B
SEND PACKET CALL EXPELLED INDICATION(S) TO PRCH MANAGER
824
GOTO C

FIG.8C

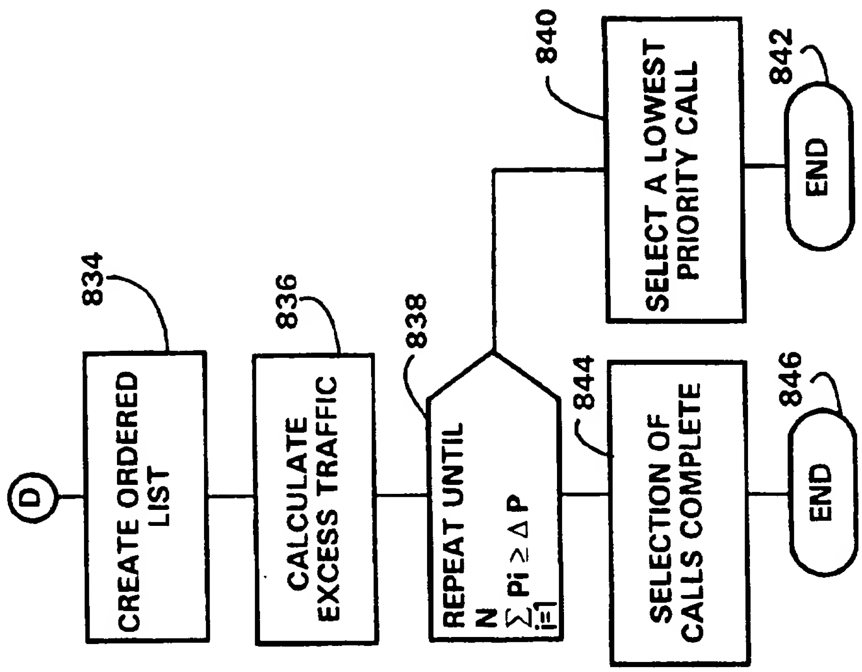

D
CREATE ORDERED LIST
834
CALCULATE EXCESS TRAFFIC
836
REPEAT UNTIL $\sum_{i=1}^{N} Pi \geq \Delta P$
838
SELECT A LOWEST PRIORITY CALL
840
END
842
SELECTION OF CALLS COMPLETE
844
END
846

FIG.8B

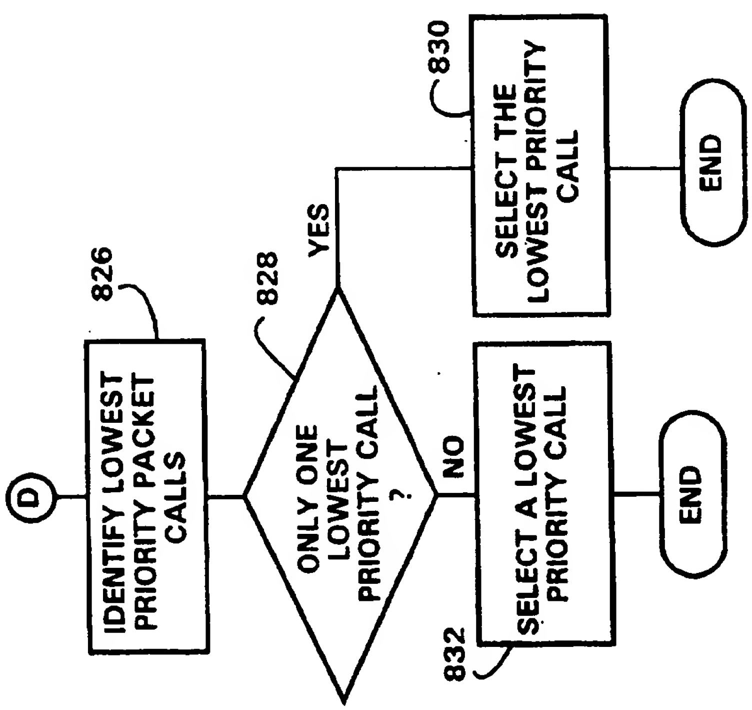

D
IDENTIFY LOWEST PRIORITY PACKET CALLS
826
ONLY ONE LOWEST PRIORITY CALL ?
828
YES
SELECT THE LOWEST PRIORITY CALL
830
END
NO
SELECT A LOWEST PRIORITY CALL
832
END

FIG.9

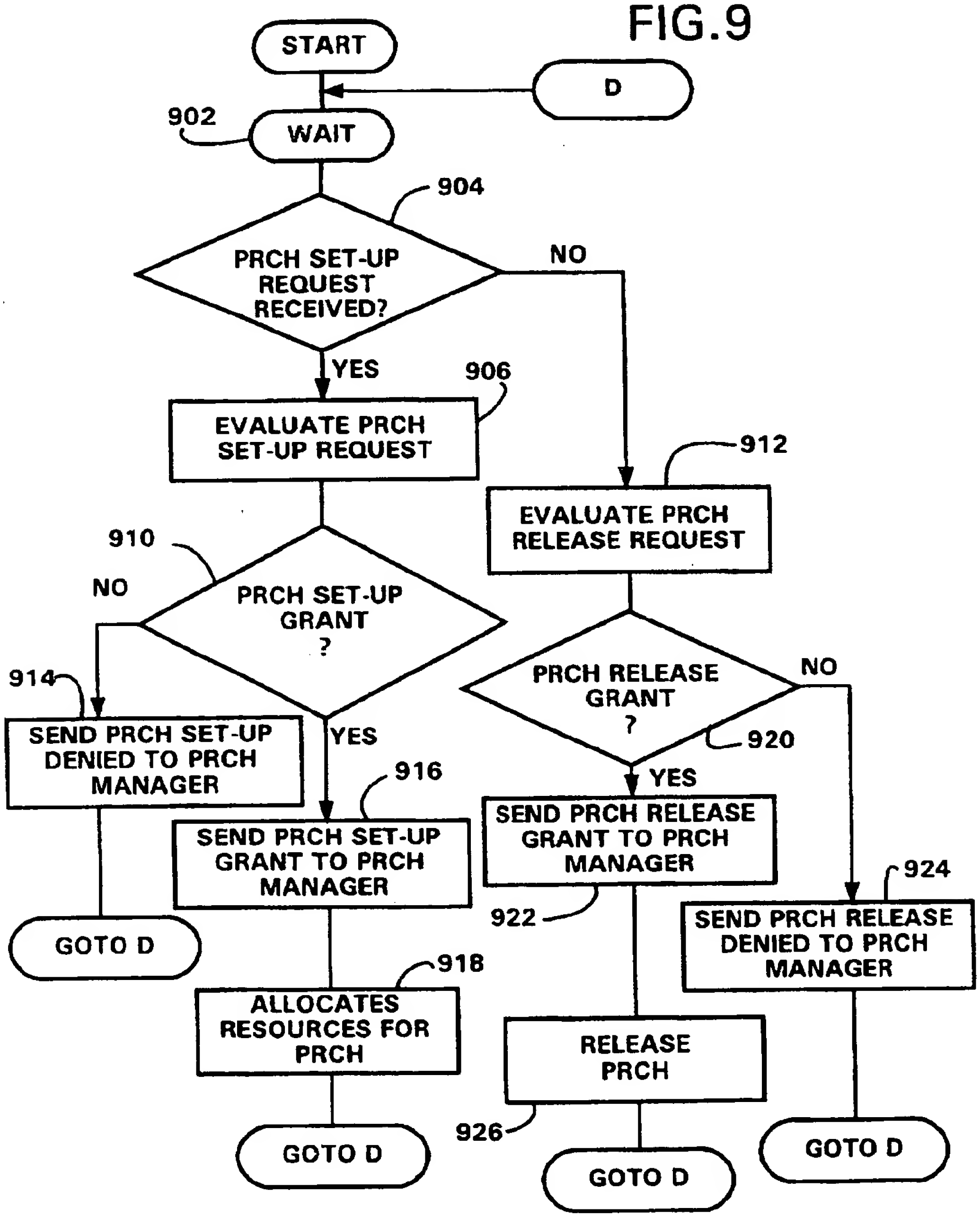

START
D
902
WAIT
904
PRCH SET-UP REQUEST RECEIVED?
NO
YES
906
EVALUATE PRCH SET-UP REQUEST
912
EVALUATE PRCH RELEASE REQUEST
910
NO
PRCH SET-UP GRANT ?
PRCH RELEASE GRANT ?
NO
914
YES
920
SEND PRCH SET-UP DENIED TO PRCH MANAGER
916
YES
SEND PRCH RELEASE GRANT TO PRCH MANAGER
SEND PRCH SET-UP GRANT TO PRCH MANAGER
924
922
SEND PRCH RELEASE DENIED TO PRCH MANAGER
GOTO D
918
ALLOCATES RESOURCES FOR PRCH
RELEASE PRCH
926
GOTO D
GOTO D
GOTO D

FIG.10

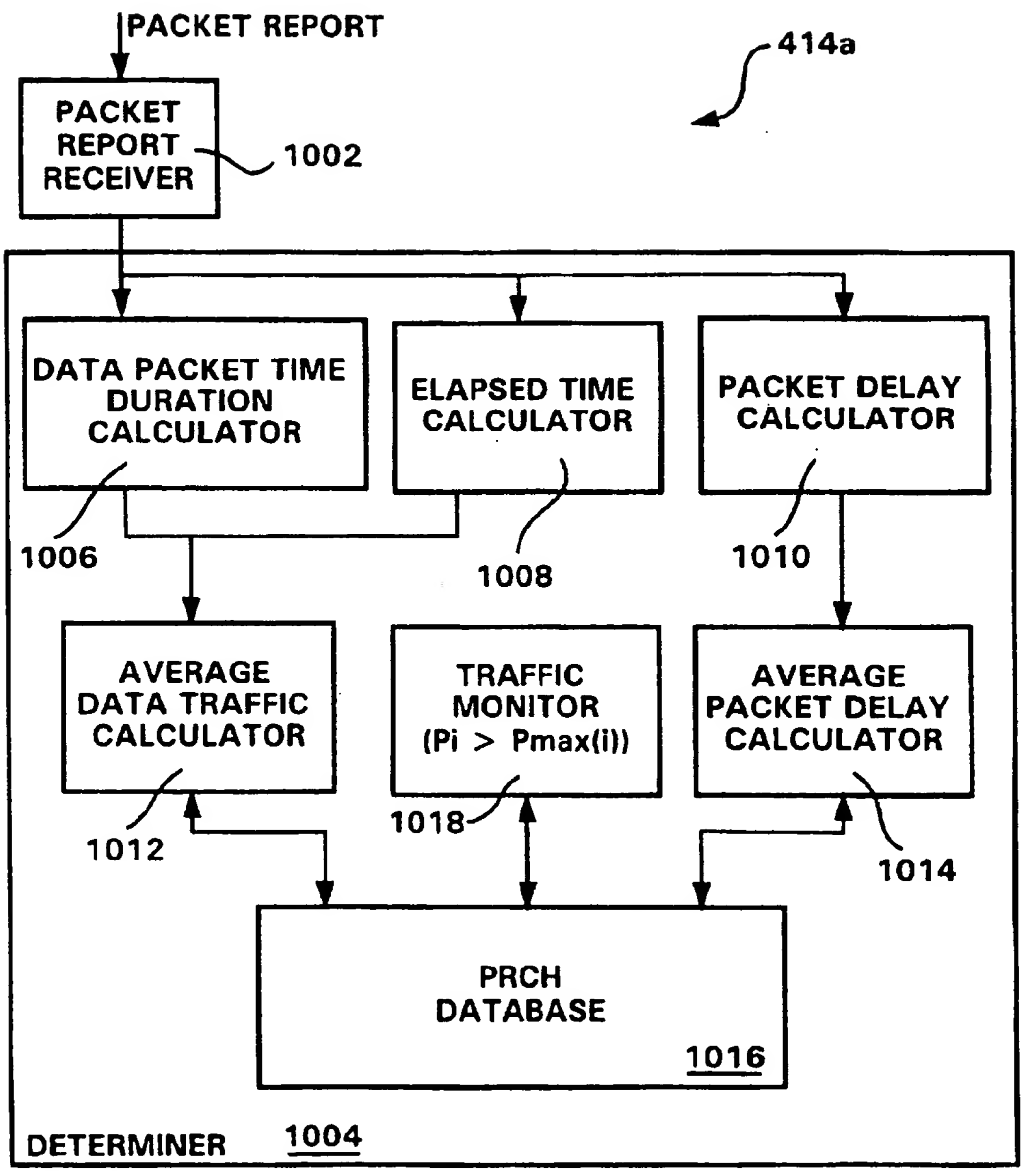
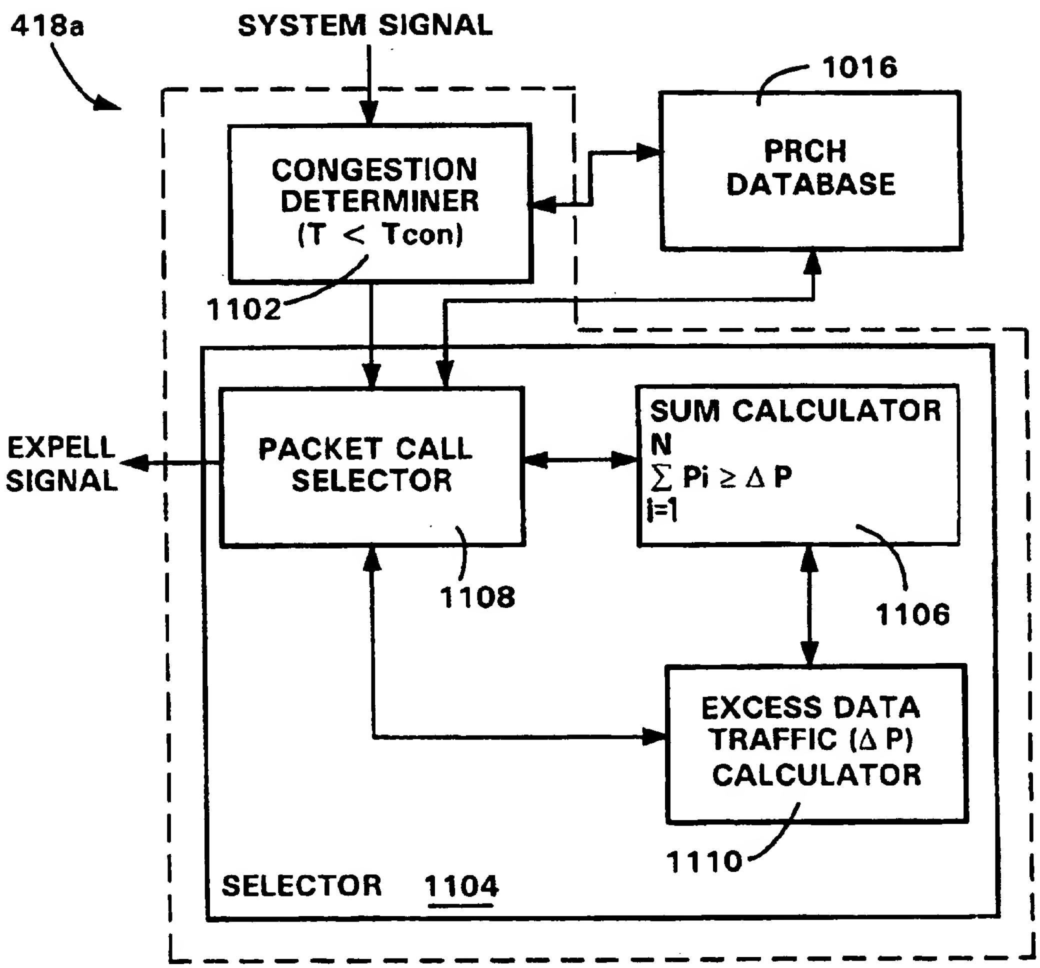

PACKET REPORT
414a
PACKET REPORT RECEIVER
1002
DATA PACKET TIME DURATION CALCULATOR
ELAPSED TIME CALCULATOR
PACKET DELAY CALCULATOR
1006
1008
1010
AVERAGE DATA TRAFFIC CALCULATOR
TRAFFIC MONITOR (Pi > Pmax(i))
AVERAGE PACKET DELAY CALCULATOR
1012
1018
1014
PRCH DATABASE
1016
DETERMINER
1004

FIG. 11



418a
SYSTEM SIGNAL
1016
CONGESTION DETERMINER (T < Tcon)
PRCH DATABASE
1102
EXPELL SIGNAL
PACKET CALL SELECTOR
SUM CALCULATOR
$\sum_{i=1}^{N} Pi \geq \Delta P$
1108
1106
EXCESS DATA TRAFFIC (Δ P) CALCULATOR
1110
SELECTOR 1104

PACKET SWITCHED RADIO CHANNEL TRAFFIC SUPERVISION

BACKGROUND OF THE INVENTION

The present application is a continuation-in-part of U.S. patent application Ser. No. 08/529,559 entitled "PACKET SWITCHED TRAFFIC MANAGEMENT IN A CELLULAR TELECOMMUNICATIONS SYSTEM", filed on Sep. 18, 1995.

FIELD OF THE INVENTION

This invention relates to packet switched telecommunications systems and, more particularly, to a method and system for packet switched radio channel traffic supervision in a telecommunications system.

HISTORY OF THE PRIOR ART

As the capability to offer a greater number and variety of services within cellular telecommunications systems develops, packet switched services will play an increasingly important role in the field of cellular telecommunications. The application of many computer and related data services to cellular systems requires the transfer of single or multiple data packets over the radio link of a cellular telecommunications system. Certain of these services such as e-mail and tele-banking may be implemented with a store and forward short message service. Other services, such as terminal emulation, local area networks, bank server access, and credit card verification, however, require interactive usage, short time delays and the capability to handle data packets of widely varying lengths. It is certain that future cellular systems will have to support such services with an efficient packet-data service.

Recognition of the importance of packet data services has resulted in the current effort of the European Technical Standards Institute (ETSI) to develop such a service for the European 2+ Group Special Mobile (GSM) cellular system. This recognition has also resulted in an effort to design packet-data service capability into the Universal Mobile Telephone System (UMTS) currently under development in the RACE II Code Division Testbed (CODIT) project R2020. The CODIT project was set up by the Commission of the European Community for the purpose of defining a future mobile telecommunications system using code division multiple access (CDMA) techniques.

Packet-switched data service in a cellular telecommunications network is characterized by calls from network users to mobile users being transmitted to packet switched mobile stations on the shared downlink (DL) of a packet switched radio channel (PRCH) and, by one or more mobile users sharing the uplink (UL) of the PRCH. The DL PRCH is shared by network users on a queued basis. The UL PRCH is shared by each mobile user accessing the channel in random fashion, as the mobile user requires, to transmit data to the system.

A common method of allowing access to the PRCH is through a packet-switched contention mode. The currently defined CODIT UMTS packet-data service is of the contention mode type. In the packet-switched contention mode mobile users transmit data packets on the PRCH when it is necessary to transfer data. An identification of the transmitting mobile user is contained in each data packet. The transmission of data packets by the mobile user may be done either randomly, or upon sensing an idle signal indicating that the packet-data channel is not presently used by another mobile station. If two or more mobile users simultaneously contend for an idle packet-data channel, the system will only allow one access to the channel. Mobile users unsuccessful at accessing the channel must repeat the transmission of the data packet until it is accepted by the system. The system users transmitting data packets to mobile users also contend for the downlink by being placed in a queue.

Because in such a system each user accesses the packet-switched channel in a random fashion, uncontrolled flow of users to, from, and between the packet-switched radio channels of a cellular system may cause packet transmission delays in the system. The delay may be incurred by both mobile users on the uplink and network users transmitting to mobile users on the downlink. As the number of packet calls on the packet switched channel increases, the average transmission delay for each packet call increases. In some applications the delays may be unacceptable.

Therefore, a need exists for a method and system for controlling packet transmission delay on one or more packet switched radio channels of a cellular system. If contending packet calls could be selectively chosen for admission to a packet radio channel according to predefined criteria, delays for packet switched channel users in applications that cannot tolerate a long packet delay time could be avoided and reduced.

A method and system for managing the flow of prioritized users to, from, and between one or more packet switched radio channels, with each packet switched radio channel having a maximum tolerable packet transmission delay, would meet such a need.

SUMMARY OF THE INVENTION

The present invention provides a method and system for packet switched radio channel (PRCH) traffic supervision. The invention allows a system operator to set the maximum average packet transmission time delay that will be incurred in a packet call. By setting a maximum average time delay on one or more PRCHs of a system and monitoring the delays in packet calls on the PRCHs, a system operator can assure that PRCH users are not subject to unacceptable delays. This avoids the problems associated with conventional contention mode packet switched systems in which users randomly contend for use of a PRCH. In such conventional systems the average time delay for packets increases as the number of users contending for the PRCH increases.

In one embodiment, the invention comprises a PRCH traffic supervision function for each PRCH of a telecommunications system. The traffic supervision function utilizes data contained in packet reports generated within the system for each data packet transmission on the PRCH. Upon receiving a new packet report the traffic supervision function calculates a packet size value (in time), a packet delay value and a value of elapsed time since the previous packet report was received. The packet size, packet delay and elapsed time are then used to calculate an estimate of average data traffic for each packet call, an estimate of average data traffic for the PRCH and an estimate of the average packet delay for the PRCH. The values calculated in the traffic supervision function may then be used to determine if a packet call should be admitted to the PRCH or if a packet call should be expelled from the PRCH when traffic on the PRCH becomes congested.

The PRCH traffic supervision function also includes an excess traffic monitor. The excess traffic monitor monitors the estimate of average data traffic for each packet call on the

PRCH to determine if the average data traffic has exceeded a required maximum data traffic for that packet call. If the average data traffic exceeds the required maximum data traffic for any packet calls, the packet calls may be expelled from the PRCH.

In alternatives of the embodiment, the estimated average data traffic for each packet call, the estimated average data traffic for the PRCH and, the average packet delay for the PRCH may be calculated for the uplink and downlink of the PRCH separately, or, as values for the combined uplink and downlink of the PRCH. The excess traffic monitor may also monitor traffic on the uplink and downlink of the PRCH separately, or, on the combined uplink and downlink of the PRCH.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and system of the present invention may be had by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

FIG. 1 is a block diagram of a cellular telecommunications system into which the present invention may be implemented;

FIG. 2 illustrates the control plane protocol architecture for the packet switching functions of a cellular telecommunications system into which the present invention may be implemented;

FIGS. 3A and 3B illustrate the exchange of signals on the downlink and uplink, respectively, of a cellular system packet radio channel operating according to an embodiment to the present invention;

FIG. 4 is a functional block diagram of packet radio traffic management functions within a cellular system operating according to an embodiment of the present invention;

FIGS. 5A–5D are flow diagrams illustrating process steps followed by the packet radio channel management function according to an embodiment of the present invention;

FIG. 6 is a flow diagram illustrating process steps followed by the packet radio channel controller traffic supervision function according to an embodiment of the present invention;

FIG. 7 is a flow diagram illustrating process steps followed by the packet radio channel controller admission control function according to an embodiment of the present invention;

FIGS. 8A–8C are flow diagrams illustrating process steps followed by the packet radio channel controller congestion control function according to an embodiment of the present invention;

FIG. 9 is a flow diagram illustrating process steps followed by the packet radio channel resource manager according to an embodiment of the present invention;

FIG. 10 is a schematic block diagram illustrating a packet traffic supervisor according to an embodiment of the present invention; and

FIG. 11 is a schematic block diagram illustrating a packet traffic congestion control function according to an embodiment of the present invention.

DETAILED DESCRIPTION

Referring now to FIG. 1, therein is illustrated a block diagram of a cellular telecommunications system 100 into which the present invention may be implemented. Cellular system 100 comprises mobile control node (MCN) 102, radio network controllers (RNCs) 104 and 106, base stations (BSs) 108, 110, 112, 114, 116 and 118, and mobile stations (MSs) 120, 122 and 124. Each base station 108, 110, 112, 114, 116 and 118 controls system radio communications with mobile stations within the radio coverage area, termed a cell, of the base station.

Mobile stations 120, 122 and 124 communicate with a particular base station, of base stations 108, 110, 112, 114, 116 and 118, depending on which base station's coverage area the mobile is located. In FIG. 1 mobile stations 120, 122, and 124 are shown to be communicating via radio interfaces 128, 130 and 132 with base stations 108, 112 and 116, respectively. Base stations 108, 110 and 112 are connected to radio network controller 104, and, base stations 114, 116 and 118 are connected to radio network controller 106. Radio network controllers 104 and 106 are in turn connected to mobile control node 102. Mobile control node 102 is a switching center that supports the interconnection of the cellular system to fixed network 126. Mobile control node 102 may be connected to fixed network 126 by landlines or other equivalent connections. The fixed network 126 may comprise an internet network, a public switched telephone network (PSTN), an integrated services digital network (ISDN), a packet switched public data network (PSPDN), or a X.25 system. While the cellular telecommunications system of FIG. 1 is shown as a particular configuration, the block diagram is intended to be only an exemplary configuration of a system into which the present invention may be implemented. The invention has application to any packet switched radio system in which users contend for a packet switched radio channel (PRCH).

In an embodiment of the invention, cellular system 100 operates according to protocols developed for the Code Division Testbed (CODIT) Universal Mobile Telephone System (UMTS) project with the PRCH contention mode access specified for the CODIT/UMTS controlled by the PRCH traffic management function of the invention. The UMTS is a mobile communication system using direct sequence code division multiple access (DS-CDMA) with a multi-rate radio interface architecture. In the CODIT/UMTS system packet radio service is provided to mobile stations 120, 122 and 124 via one or more PRCHs. Each base station 108, 110, 112, 114, 116 and 118 establishes and terminates one or more PRCHs at the request of radio network controllers 104 and 106 or mobile control node 102. The PRCH is a full duplex, asymmetrical channel that can be operated independently on both the uplink (UL) and downlink (DL) at variable mobile station data rates up to 9.6 kbps (narrow band channel) or up to 64 kbps (medium band channel). MCN 102 can attach multiple mobile stations to a single PRCH within a single cell. To distinguish several mobile stations on a PRCH, MCN 102 assigns each mobile a virtual connection identifier (VCI) when it grants access. The VCI is represented by a k bit number and serves as a unique address within the area controlled by MCN 102.

The PRCH is structured in ten ms time slots to convey fragmented packets between mobile stations 120, 122 and 124 and the network. On the DL, the mobile control node 102 can send mobile station data packets and information for controlling the access and data transfer on the UL to one mobile station or simultaneously to a plurality of mobile stations. On the UL, the mobile stations may share access to a UL PRCH if within the coverage area of the same base station. After gaining access to the PRCH, the mobile station transmits the packet to the system over a physical channel. The logical channel PRCH is mapped onto two physical channels comprising a physical data channel (PDCH) and a

physical control channel (PCCH). Two base station transceivers are required for supporting one PRCH.

Referring now to FIG. 2, therein is illustrated the protocol stack 200 for the packet switching functions of the CODIT/UMTS. In the mobile station the mobile station protocol stack (MS/PS) 218 comprises a network layer 202, data-link control (DLC) layer 204, a medium access control (MAC) layer 206, and the physical layer 208. On the network side, the network protocol stack (NW/PS) 220 comprises a network layer 210 and a DLC layer 212, each located within either the MCN or RNC, a medium access layer (MAC) 214 located within the base station and MCN or RNC, and a physical layer 216.

The connectionless packet service (CLPS) entity of network layer 202 provides the packet service to the mobile station. The CLPS of network layer 210 provides the functions of registration, authentication, assigning and administering VCIs and interfacing to a packet data network. During a packet call, the CLPS entities use a logical link administrator (LLA) to initially route packet service set-up signals via a dedicated control channel (DCCH and CC). After the packet service set-up, the mobile station is attached to a PRCH and all messages between the CLPS, including mobile station data packets, are passed through the DLC to a packet-radio (PR) control entity. The PR entity is also responsible for normal mobile telephone system functions such as handover, connection re-establishment, etc.

The packets to be transmitted on the PRCH are fragmented, protected with a block code (BC) for detecting transmission errors on the receiving side, convolutionally encoded, interleaved (IL), switched through a multiplexer (MUX) and then transmitted over the PDCH. Control information, e.g. for power control, may also be transferred via the PCCH. On the receiving side, the fragments are reconstructed from the received samples, reassembled into packets, and forwarded to a connectionless packet service (CLPS) entity. When a block decoder on the receiving side detects the receipt of an erroneous packet fragment, a packet radio control function requests its retransmission. In cellular system 100 there may be several PRCHs distributed among the cells controlled by base stations 108, 110, 112, 114, 116 and 118.

Referring now to FIGS. 3A and 3B, therein are illustrated the exchange of signals on the uplink (UL) and downlink (DL), respectively, of a cellular system PRCH operating according to the present invention. FIGS. 3A and 3B show the signal exchanges between a mobile station (MS) 300 and the network (NW) 302. Mobile station 300 is shown functionally as mobile station protocol stack (MS/PS) 218 and mobile station system manager (MS/SM) 220. Network 302 is shown functionally as network protocol stack (NW/PS) 222 and network system manager (NW/SM) 224. The protocol stack is responsible for data transmission and the system manager is responsible for control and supervision of the connection between the network and the mobile station.

For uplink (UL) packet transmission and reception the following scheme is used (the steps correspond to the numbering of the arrows in FIG. 3A).

1U. The MS/PS 218 can send three different kinds of packets to the NW/PS 222, two of which require acknowledgment.

- a. Packets requiring acknowledgment:
  - packets containing user data; and
  - packets containing user data with piggy-backed downlink reports (DLRs).
- b. Packets not requiring acknowledgment:
  - packets containing only DLRs.

A timer is set in MS/SM 220 when a packet requiring acknowledgment is sent. If the timer expires before an acknowledgment is received, the packet is considered to be lost.

2U. For all UL data packets, quality samples are sent to NW/SM 224. At the end of the UL packet a packet stop signal is sent to the NW/SM 224 indicating that the last quality sample has been sent for that particular packet.

3U. After receiving a UL data packet, a UL packet report is sent to NW/SM 224. This report contains information required for traffic supervision.

4U. If the UL packet contains a piggy-backed DLR or if the packet is a stand-alone DLR, the DL quality estimate is extracted and forwarded to NW/SM 224.

5U. If the transmitted UL data packet requires an acknowledgment, an acknowledgment message is sent from NW/PS 222 to the MS/PS 218. The message can be either stand-alone or piggy-backed on a DL mobile station information packet.

6U. Upon receiving an acknowledgment in MS/PS 218, A packet acknowledged signal is sent to MS/SM 220. If no acknowledgment is received before the timer introduced in Step 1 above expires, a packet lost message is sent to the MS/SM 220.

For DL packet transmission and reception the following scheme is used (the steps correspond to the numbering of the arrows in FIG. 3B):

1D. The NW/PS 222 can send three different kinds of packets to the MS/PS 218, two of which require acknowledgment.

- a. Packets requiring acknowledgment:
  - packets containing user data; and
  - packets containing user data with piggy-backed acknowledgment/no acknowledgment (ack/nack) information for previously received UL packets.
- b. Packets not requiring acknowledgment:
  - packets containing only ack/nack information for previously received UL packets.

A timer is set when packets requiring acknowledgment are sent. If the timer expires before an acknowledgment is received, the packet is considered to be lost.

2D. When a DL data packet is transmitted, a DL packet report is sent to NW/SM 224. The report contains information required for traffic supervision.

3D. When receiving a DL data packet in MS/PS 218, quality samples are extracted for each frame and sent to MS/SM 220. At the end of the DL packet, a packet stop signal is sent to MS/SM 220 indicating that the last quality sample has been sent for that particular packet.

4D. After receiving a packet stop signal, a quality estimate is sent to MS/PS 218. This estimate is a measure of the quality of the entire packet sent on the DL.

5D. A DownLink Report (DLR) containing an ack/nack message and a quality estimate is sent to NW/PS 222 for each received DL packet containing user data. The DLR can be sent either stand-alone or piggy-backed on a UL user data packet. After receiving the DLR in NW/PS 222, the quality estimate is forwarded to NW/SM 224.

6D. If the ack/nack information in the DLR contains an acknowledgment, a packet acknowledged signal is sent to NW/SM 224. If no acknowledgment is received before the timer introduced in Step 1 above expires, a packet lost message is sent to the NW/SM 224.

Referring now to FIG. 4, therein is a functional block diagram of packet radio traffic management functions within a cellular system operating according to the present inven-

tion. The functionality of the packet radio traffic management, which is logically located in the NW/SM **224**, comprises three main blocks: PRCH manager **402**, resource manager **404** and PRCH controllers **406***a*, **406***b*, **406***c* and **406***d*. Normally, there is one PRCH manager **402** for each base station of the system. If a base station supports more than one cell, there is one PRCH manager **402** for each cell. The number of PRCH controllers **406***a*, **406***b*, **406***c* and **406***d*, depends on the number of PRCHs necessary and, resources available, for packet switched traffic in the cell. In the embodiment shown in FIG. **4**, there are four PRCHs in the cell. Each PRCH controller controls one PRCH comprising an uplink and downlink. The PRCH manager **402** is invoked when it is necessary for a user to have access to a PRCH of the cell. Reception of a service request via the NW/PS **222** invokes the PRCH manager **402**. The PRCH manager **402** will also be invoked if a packet call has been expelled from a PRCH due to congestion and a packet call expelled indication is received from a PRCH controller. Additionally, the PRCH manager **402** will be invoked if an internally generated admission queue signal or a PRCH setup grant/denial or release grant/denial signal from the resource manager is received.

A service request could be received in any of the following situations:

1) A new user wants access to a PRCH to initiate packet switching service.

2) A user wants to make handover from a PRCH of another cell to a PRCH of the cell in which PRCH manager **402** is located.

3) A user wants to re-establish a lost PRCH connection.

4) A user wants to update its traffic requirements, see below.

Each traffic event listed above results in a service request being forwarded to the PRCH manager. The service request contains information necessary for evaluation by service request evaluation function **408** of PRCH manager **402**. The information includes:

Type of request

Required estimated average user data traffic, P_{ave} (scaled to the maximum user bi trate of the PRCH). This comprises separate parameters for each of the UL and DL.

Required estimated maximum user data traffic, P_{max} (scaled to the maximum user bi trate of the PRCH). This comprises separate parameters for each of the UL and DL.

Priority, Pri. This parameter can assume a value within the interval $[0,Pri_{max}]$. The priority can be assigned on the basis of the mobile station initiating the call or being called, or on another basis.

A service request is evaluated through the service request evaluation function **408**. In the service request evaluation, the PRCH manager **402** sends a PRCH admission request for a packet call to one of PRCH controllers **406***a*, **406***b*, **406***c*, or **406***d*. PRCH manager **402** will try each PRCH controller **406***a*, **406***b*, **406***c*, or **406***d* until admission is granted or the packet call is not admitted in any of the PRCHS. If the packet call is not admitted in any of the existing PRCHs (the PRCH admission request is denied by all PRCH controllers **406***a*, **406***b*, **406***c*, and **406***d*). PRCH manager **402** decides if the service request should be denied or if the packet call should be put in the admission queue **420** by using the admission queue handling function **410**.

A packet call placed in the admission queue is temporarily suspended, i.e., no information is allowed to be exchanged between the users. If the packet call is not placed in the admission queue, a service denied signal is sent to the user. If the packet call is to be placed in the admission queue, the PRCH manager informs the users by sending a packet call suspend indication signal.

A packet call expelled indication signal is received in PRCH manager **402** from a PRCH controller when a packet call is expelled from a PRCH due to congestion, i.e., the packet call is removed from the PRCH. A packet call expelled indication signal is evaluated through the packet call expelled evaluation function **422**. In the packet call expelled evaluation function **422** the PRCH manager **402** sends a PRCH admission request for the expelled packet call to one of PRCH controllers **406***a*, **406***b*, **406***c* or **406***d*. PRCH manager **402** will try each PRCH controller **406***a*, **406***b*, **406***c* or **406***d* until admission is granted or the expelled packet call is not admitted in any of the PRCHS.

If the packet call is not admitted in any of the existing PRCHs, PRCH manager **402** decides if the expelled packet call should be detached or if the expelled packet call should be put in the admission queue **420** by using the admission queue handling function. If the expelled packet call is placed in admission queue **420**, the packet call is temporarily suspended and a packet call suspend indication signal is sent to the user via NW/PS **222**. If the expelled packet call is not placed in the admission queue **420**, a packet call detach indication signal is sent to the user via NW/PS **222**.

A packet call admission queue signal indicates that the admission queue **420** should be checked. The admission queue signal may be generated by a timer set as the system operator desires. A packet call admission queue signal is evaluated through the admission queue handling function **410**. In the admission queue handling function the PRCH manager **402** sends a PRCH admission request for the packet call in the admission queue with the highest priority to one of PRCH controllers **406***a*, **406***b*, **406***c* or **406***d*. PRCH manager **402** will send the admission request to each PRCH controller **406***a*, **406***b*, **406***c* or **406***d* until admission is granted or the packet call is not admitted in any of the PRCHs. If the packet call is admitted to any of the PRCHs, a packet call resume indication signal is sent to the user via NW/PS **222**.

PRCH manager **402** also decides when it is necessary to set up a new PRCH or release an existing PRCH through the PRCH management function **412**. In the case of both PRCH setup and PRCH release, a step up or release request signal is sent to resource manager **404** which controls the allocation of system resources for PRCHs. Resource manager **404** either denies or grants the request by sending a setup request grant or a setup request denied signal to PRCH manager **402** or sending a release request grant or release request denied signal to PRCH manager **402**.

Each PRCH controller **406***a*, **406***b*, **406***c* and **406***d* supervises the traffic on one PRCH of the cell. There is one PRCH controller for each PRCH in a cell. Each PRCH controller **406***a*, **406***b*, **406***c* and **406***d* receives traffic information on the PRCH that it controls from NW/PS **222** in a packet report. The packet report is evaluated by the PRCH traffic supervision function, **414***a*, **414***b*, **414***c* or **414***d*, for the relevant PRCH. The information contained in the packet report is used to decide if new packet calls can be admitted to the PRCH through the PRCH admission control function, **416***a*, **416***b*, **416***c* or **416***d*, when an admission request is received from PRCH manager **402**. The information contained in the packet report may also be used to decide if the PRCH congestion control function, **418***a*, **418***b*, **418***c* or **418***d*, should be used to expel an already admitted packet

call due to PRCH overload. In this case a packet call expelled indication signal is sent to the PRCH manager. The PRCH manager then decides if the packet call should be temporarily suspended or detached through the packet call expelled evaluation function **422**. Depending on this decision, the users are informed by a packet call suspend indication signal or a packet call detach indication signal.

Resource manager **404** controls the allocation of system resources for packet radio channels. The PRCH manager **402** may request that a new PRCH be set up or released by sending a PRCH setup/release request to resource manager **404**. The PRCH manager **404** continuously monitors the size of the admission queue **420**. Whenever the total required estimated average data traffic of all packet calls in the admission queue P_q exceeds a limit P_{new} PRCH set for the admission queue, a PRCH setup request is sent to the higher level resource manager **404**. If P_{new} PRCH is set to zero, the PRCH manager always requests more resources as soon as the existing PRCHs are full. As soon as the number of users attached to a PRCH is zero, a PRCH release request is sent to the resource manager **404**. If granted, the PRCH is released.

PRCH manager **402** and PRCH controllers **406***a*, **406***b*, **406***c* and **406***d* may be implemented into the base stations, radio network controllers and mobile control nodes of a cellular system such as the system shown in FIG. **1**. The actual implementation may be in either hardware or software, or in a combination of hardware and software, operating in conjunction with one or more processors. Processors and software for implementing these types of functions are well known in the art.

Referring now to FIGS. **5**A, **5**B, **5**C and **5**D, therein are shown traffic flow diagrams illustrating service request evaluation, packet call expelled evaluation, admission queue handling and PRCH management process steps, respectively, followed by PRCH manager **402** according to an embodiment of the present invention.

The PRCH manager **402** receives an input while in the wait state of Step **502** of FIG. **5**A. The input may be a service request, a packet call expelled indication, an internally generated admission queue signal or, a PRCH setup grant or denied signal or release grant or denied signal received from resource manager **404**. At Step **504** it is determined if a service request was received from NW/PS **222**. If a service request was not received, the process moves to Step **534** of FIG. **5**B. If, however, a service request was received, the process moves to Step **506** and begins the service request evaluation.

The service request evaluation of Step **506** involves requesting PRCH admission in Steps **508**, **510**, **512**, **514**, **516**, **518** and **520**. The service request evaluation is repeated for each PRCH controller **406***a*, **406***b*, **406***c* and **406***d*, sequentially, until admission to a PRCH is granted or no PRCHs remain. At Step **508** PRCH manager **402** sends a PRCH admission request to one of PRCH controllers **406***a*, **406***b*, **406***c* or **406***d*. The process then moves to Step **510** as PRCH manager **402** waits for a response. The PRCH manager **402** periodically checks at Step **512** to determine whether a response has been received from PRCH controllers **406***a*, **406***b*, **406***c* or **406***d*. If no response has been received, the process moves back to the wait state of **510**. If, however, it is determined at Step **512** that a response has been received from PRCH controller **406***a*, **406***b*, **406***c* or **406***d*, the PRCH admission request process is completed and the process moves to Step **514**, where it is determined if the response is an admission grant. If the response is an admission grant, the service request evaluation process is completed at Step **520** and the process moves to Step **522**.

If, however, at Step **514**, it is determined that the response is not an admission grant, it is an admission denied response, and the process moves to Step **516** where it is determined if the current response was sent from the last PRCH controller to which an admission request could be sent. If it was not the last PRCH controller, the process moves to Step **518** and continues the service request evaluation process of Step **506** for the next PRCH. The service request evaluation process of Step **506** is repeated until an admission grant response is received from PRCH controller **406***a*, **406***b*, **406***c* or **406***d*, or, until all PRCH controllers have denied admission. When the service request evaluation process is completed the process moves to Step **522**.

At Step **522** it is determined if an admission grant response was received from any PRCH controller. If an admission grant was received from a PRCH controller, the process moves to Step **524** where a service grant signal is sent to the user via the NW/PS **308**. From Step **524** the process then moves to Step **534** of FIG. **5**B. If, however, at Step **522** it is determined that no admission grant was received from any PRCH controller the process moves to Step **528**. At Step **528** PRCH manager **402** determines, using the admission queue handling function **410**, if the packet call is to be put in the PRCH admission queue. It is determined to put the packet call in the admission queue **420** if the following criterion is fulfilled:

$$P_{ave}(r)+P_q(r)<P_{max}(r)$$

$P_{ave}(r)$ is the required estimated average data traffic for the user as a function of the service request r and $P_q(r)$ is the requested traffic of all packet calls in the admission queue of service request type r. $P_q(r)$ is a measure of the current size of the queue for the service request type. $P_{max}(r)$ is the maximum allowed requested traffic in the admission queue **420** as a function of the service request. In alternatives of the embodiment, the comparison may be done using $P_{ave}(r)$, $P_q(r)$ and $P_{max}(r)$ values for the uplinks and downlinks separately, or, using values for the uplinks and downlinks combined. It is possible to have a different P_{max} for different types of service requests, r. Thereby a prioritization between different service requests can be done in Step **528**. For example, when requesting a PRCH during handoff, the value of $P_{max}(r)$ may be set higher than the value of $P_{max}(r)$ is set when requesting access to a PRCH for the first time.

If it is determined, at Step **528**, that the packet call is to be put in the PRCH admission queue, the call identity is placed in the admission queue **420** and the process moves to Step **531** where a service grant signal is sent to the user via NW/PS **222**. The process next moves to Step **532** where a packet call suspend indication signal is sent to the user via the NW/PS **308**. The process then moves to Step **534** of FIG. **5**B. If, however, at Step **528**, it is determined that the packet call is not to be put in the PRCH admission queue **420** the process moves to Step **530** and a service denied signal **428** is sent to the user. The process then moves to Step **534** of FIG. **5**B.

At Step **534** of FIG. **5**B, it is determined if a packet call expelled indication was received. If the input was not a packet call expelled indication, the process moves to Step **562** of FIG. **5**C. If, however, it is determined at Step **534** that a packet call expelled indication was received, the process moves to Step **536**. At Step **536** a PRCH admission request for the expelled packet call is sent to PRCH controller **406***a*, **406***b*, **406***c* or **406***d* from PRCH manager **402**. The admission request process of Step **536** involves Steps **538**, **540**, **542**, **544**, **546**, **548** and **550**. Step **536** is repeated for each PRCH controller **406***a*, **406***b*, **406***c* or **406***d* until admission

has been requested to all PRCHs. At Step **538** PRCH manager **402** sends a PRCH admission request to PRCH controller **406a**, **406b**, **406c** or **406d**. The process then moves to Step **540** as PRCH manager **402** waits for a response. The PRCH manager **402** periodically checks at Step **542** to determine whether a response has been received from PRCH controller **406**. If no response has been received, the process moves back to the wait state of Step **540**. If, however, it is determined at Step **542** that a response has been received from the PRCH controller to which the admission request has been sent, the process moves to Step **544** where it is determined if the response if an admission grant. If the response is an admission grant, the packet call expelled evaluation ends at Step **550** and the process moves to Step **552**. If, however, at Step **544**, it is determined that the response is not an admission grant, it is an admission denied response and the process moves to Step **546** where it is determined if the admission denied response was sent from the last PRCH controller to which an admission request could be sent. If it was not the last PRCH controller, the process moves to Step **566** and repeats the admission request process of Step **536** for the next PRCH. The packet call expelled evaluation of Step **536** is repeated until an admission grant response is received from a PRCH controller or, until all PRCH controllers **406a**, **406b**, **406c** and **406d** have denied admission. When the packet call expelled evaluation process of Step **536** is completed, the process moves to Step **552**.

At Step **552** it is determined if an admission grant response was received from any PRCH controller during Step **536**. If an admission grant was received from a PRCH controller, the process moves to Step **554** where a packet call update indication signal is sent to the user via the NW/PS **222**. From Step **554** the process moves to Step **562** of FIG. 5C. If, however, at Step **552** it is determined that an admission grant was not received, the process moves to Step **556**. At Step **556** PRCH manager **402** determines, using the admission queue handling function **410**, if the expelled packet call is to be put in the PRCH admission queue. The same admission criteria are used at Step **556** as was described for Step **528** of FIG. 5A. If it is determined at Step **556** to place the expelled packet call in the admission queue **420**, the process moves to Step **560** and a packet call suspend indication signal is sent to the user via NW/PS **222**. The process then moves from Step **560** to Step **562** of FIG. 5C. If, however, it is determined at Step **556** not to place the expelled packet call in the admission queue **420**, the process moves to Step **558** and a packet call detach indication signal is sent to the user via NW/PS **222**. The process then moves from step **558** to Step **562** of FIG. 5C.

At Step **562** of FIG. 5C it is determined if an admission queue signal was received. If an admission queue signal was not received, the process moves to Step **584** of FIG. 5D. If, however, it is determined that an admission queue signal was received, the process moves to Step **563**. At Step **563** it is determined if any packet calls are in the PRCH admission queue. If no packet calls are in the PRCH admission queue **420** of the cell, the process moves to the wait state of Step **502** in FIG. 5A. At Step **502** the process will wait for an input. If, however, it is determined at Step **563** that the PRCH admission queue **420** contains packet calls, the process moves to Step **564**. At Step **564** a PRCH admission request for the packet call having a highest priority in the admission queue **420** is sent to PRCH controller **406a**, **406b**, **406c** or **406d**, *from PRCH manager 402*.

The admission request process of Step **564** involves Steps **566**, **568**, **570**, **572**, **574**, **576** and **578**. Step **564** is repeated for each PRCH controller, **406a**, **406b**, **406c** or **406d**, until admission to a PRCH is granted or, until admission has been requested to all PRCHs. At Step **566** PRCH manager **402** sends a PRCH admission request to PRCH controller **406a**, **406b**, **406c** or **406d**. The process then moves to Step **568** as PRCH manager **402** waits for a response. The PRCH manager **402** periodically checks at Step **570** to determine whether a response has been received from PRCH controller **406**. If no response has been received, the process moves back to the wait state of **568**. If, however, it is determined at Step **570** that a response has been received from the PRCH controller to which the admission request had been sent the process moves to Step **572**, where it is determined if the response is an admission grant. If the response is an admission grant, the admission request process ends at Step **578** and the process moves to Step **586**. If, however, at Step **572**, it is determined that the response is not an admission grant, it is an admission denied response, and the process moves to Step **574** where it is determined if the admission denied response was sent from the last PRCH controller to which an admission request could be sent.

If it was not the last PRCH controller, the process moves to Step **566** and repeats the admission request process of Step **564** for the next PRCH. The admission request evaluation of Step **564** is repeated until an admission grant response is received from a PRCH controller or, until all PRCH controllers **406a**, **406b**, **406c** and **406d** have denied admission. When the admission request process of Step **564** is completed the process moves to Step **580**.

At Step **580** it is determined if an admission grant response was received from any PRCH controller in Step **564**. If an admission grant response was received from a PRCH controller, the packet call having a highest priority in the admission queue **420** is removed from the queue and the process moves to Step **582** where a packet call resume indication signal is sent to the user via the NW/PS **222**. From Step **582** the process moves to Step **584** of FIG. 5D. If, however, at Step **580** it is determined that an admission grant was not received, the process moves directly to Step **584** of FIG. 5D.

At Step **584** of FIG. 5D it is determined if a PRCH setup grant was received from resource manager **402**. If a PRCH setup grant was received from resource manager **402**, the process moves to Step **586** and the PRCH manager creates a new PRCH controller. Next, the process moves to Step **592**. If, however, at Step **584**, it is determined that a PRCH release grant was not received, the process moves to Step **588** where it is determined if a PRCH release grant was received from resource manager **402**. If a PRCH setup grant was received, the process moves to Step **590** where the PRCH manager deallocates resources from the PRCH controller for which the release request was sent. Next, the process moves to Step **592**. If, however, at Step **588**, it is determined that a PRCH setup grant was not received, the process moves directly to Step **592**.

At Step **592** the requested traffic for all packet calls in the admission queue are evaluated. Next, at Step **594**, it is determined if a new PRCH is required. If the total required estimated average data traffic of all packet calls in the admission queue P_q exceeds a limit P_{new} PRCH set for the admission queue, a new PRCH is required and the process moves to Step **596**. In alternatives of the embodiment, the comparison of P_q and P_{new} PRCH may be done using a P_q and P_{new} PRCH value for the uplinks and downlinks separately, or, using P_q and P_n PRCH values for the uplinks and downlinks of the cell combined. At Step **596** a PRCH setup request is sent to resource manager **404**. From Step

596 the process returns to the wait state of Step **502**. If, however, at Step **594** it is determined that a new PRCH is not required, the process moves to Step **597**.

At Step **597** the number of packet calls on each PRCH is evaluated. Next, at Step **598**, it is determined if any PRCH exists that is not carrying any packet calls. If it is determined that no PRCH not carrying any packet calls exists, the process returns to Step **502** of FIG. **5**A. If, however, at Step **598** it is determined that one or more PRCHs exist that are not carrying packet calls, the process moves to Step **599** where a PRCH release request is sent to resource manager **404** for each PRCH not carrying any packet call. From Step **599** the process returns to the wait state of Step **502** of FIG. 5A.

Referring now to FIGS. **6**, **7** and **8A–8C**, therein are illustrated flow diagrams showing Steps followed by each PRCH controller, **406a**, **406b**, **406c** or **406d**, for the PRCH traffic supervision, PRCH admission control and PRCH congestion control processes, respectively, according to an embodiment of the present invention. PRCH controllers **406a**, **406b**, **406c** and **406d** each continuously supervise data traffic, the average packet delay and, also receive admission requests for a PRCH.

When initially activated upon receiving an input from PRCH manager **402**, the process is in the wait state of Step **602** of FIG. **6**. While in the wait state of Step **602**, each PRCH controller **406a**, **406b**, **406c** and **406d** may receive an input in the form of a packet report from the NW/PS **222**, an admission request from PRCH manager **402** or an internally generated activation signal indicating a PRCH congestion check should be done. Upon receiving an input the process moves to Step **604** where it is determined if a packet report was received. If it is determined that a packet report was not received, the process moves directly to Step **708** of FIG. **7**. If, however, at Step **604**, it is determined that a packet report was received, the process will move to Step **606** where the PRCH traffic supervisor function **414** updates traffic statistics for the relevant PRCH. The traffic statistics are updated using information contained in the packet report. Each packet report contains the following information:

1) Transmitting mobile user identity for UL or transmitting network user identity for DL.
2) Packet size (number of frames)
3) Time stamp (indicating when the packet was placed in the transmission buffer).
4) Packet type (UL or DL). Using the information contained in the packet report the PRCH controller calculates the following:

1) The packet size (in time), X, is calculated using knowledge about the frame size.
2) The packet delay, D, is calculated as the difference between the time the packet report was received and the time the packet was placed in the transmission buffer (as indicated by the time stamp). Depending on when the packet report is sent from the protocol stack (at the beginning of the transmission or after transmission completion), the calculated delay is adjusted so that it corresponds to the time elapsed at transmission completion.
3) The time elapsed, Δt, since the previous packet report with the same packet identifier is received. The time of receival for the last packet report for each packet call is stored for this purpose.

X, D and Δt are then used to calculate an estimate of average data traffic (Pi) for each individual packet call an estimate of average data traffic (Pchan) for all packets calls on the PRCH, and an estimate of the average packet delay (T) for all packet calls on the PRCH. In alternatives of the embodiment, values of the Pi, Pchan and T may be calculated for the uplink and downlink of a PRCH separately, or as values for the combined uplink and downlink of the PRCH. The alternative used depends on which type of value the system operator needs for other functions, i.e., whether other functions in the system are using values for the uplink and downlink separately or, uplink and downlink combined.

The estimate of average data traffic Pi_N may be updated by calculating Pi for each new packet report (number N) of the packet call i as follows:

$$Pi_N = \alpha_N Pi_{N-1} + (1-\alpha_N)\frac{X_N}{\Delta t_N}$$

where

$$\alpha_N = \frac{1}{1 + e^{\frac{\Delta t_N}{\tau}}\left(\frac{\Delta t_N}{\Delta t_{N-1}}\right)(1-\alpha_{N-1})}\ ;\ \alpha_1 = 0$$

The time constant τ corresponds to the filter memory (correlation time).

In the calculation of Pi, the contribution from a single packet ($X/\Delta t_{jj}$) is weighted by the factor.

$$\Delta t_j e^{-\frac{t_j}{\tau}}$$

where t_j denotes the time elapsed since the last packet report for packet call j and Δt_j denotes the time elapsed between packet report j-1 and j. This particular weighting factor gives older samples less weight than newer samples and proportions the weight to the time period Δt_j associated with the sample.

The equations shown above for Pi calculation may also be used to calculate Pchan. In this case, the variable Pi_N and Pi_{N-1} would be replaced by $Pchan_N$ and $Pchan_{N-1}$, respectively, and packet reports from all packet calls on the PRCH would be used in the calculations.

The estimate of the average packet delay (T_N) for the PRCH may be updated by calculating T for each new packet report (number N) of the PRCH as follows:

$$T_N = \alpha_N T_{N-1} + (1-\alpha_N)D$$

where,

$$\alpha_N = \frac{1}{1 + e^{\frac{\Delta t_N}{\tau}}(1-\alpha_{N-1})}\ ;\ \alpha_1 = 0$$

The time constant τ corresponds to the filter memory (correlation time).

In the calculation of T the contribution from a single packet (T) is weighted by the factor:

$$e^{\frac{t_j}{\tau}}$$

where t_j denotes the time elapsed since the last packet report received on the PRCH. This particular weighting factor gives older samples less weight than newer samples.

The values Pi, Pchan and T may be used at Step **608** and for the admission control process (FIG. **7**), and the congestion control process (FIG. **8**).

After updating the traffic statistics at Step **606**, the process moves to Step **608**.

At Step **608** it is determined if the excess traffic monitor function is active. If a determination is made that the excess traffic monitor function is not active, the process moves to Step **708** of FIG. **7**. If, however, it is determined that the excess traffic monitor function is active, the process moves to Step **610** where it is determined if any packet call i exists on the PRCH meeting the condition, $P_i > P_{max(i)}$. If no packet calls exist on the PRCH with $P_i > P_{max(i)}$, the process moves to Step **708** of FIG. **7**. If, however, at Step **610**, it is determined that packet call(s) exist meeting the condition, $P_i > P_{max(i)}$, the process moves to Step **612**. At Step **612** the packet call or packet calls with $P_i > P_{max(i)}$ are expelled from the PRCH and a packet call expelled indication is sent to the PRCH manager **402** indicating which packet call or packet calls were expelled. The process then moves to Step **708** of FIG. **7**. As an alternative to expelling the packet on the PRCH with $P_i > P_{max(i)}$, the system could send a request to the user to change priority or increase its traffic requirements. A change in traffic requirements would result in a higher $P_{max(i)}$ for the packet call.

Referring now to FIG. **10**, therein is a schematic block diagram illustrating one hardware embodiment of the packet traffic supervision function **414a** of FIG. **4**. In the embodiment shown in FIG. **10**, the traffic supervision function comprises a packet report receiver **1002** and determiner **1004** for determining the traffic statistics. Determiner **1004** comprises a data packet time duration calculator **1006**, elapsed time calculator **1008**, packet delay calculator **1010**, average data traffic calculator **1012**, average packet delay calculator **1014**, database **1016**, and excess traffic monitor **1018**.

FIG. **7** illustrates the steps performed by packet radio channel admission control function of the invention. The flow diagram of FIG. **7** will be entered at Step **708** from Steps **604**, **608**, **610** or **612** of FIG. **6**. At Step **708** it is determined if the input was an admission request. If an admissions request was not received, the traffic statistics have been updated or an internally generated activation signal indicating that PRCH congestion check should be done has been received, and the process will move directly to Step **818** of FIG. **8**. If however, at Step **708**, it is determined that an admission request was received, the process will move to Step **710** where the admission request is evaluated.

The PRCH admission control function **416** evaluates the PRCH admission request by determining if the following is true:

$$P_{ave_N} + \Sigma P_i < P_{tol}, i \in U(Pri)$$

where,

- P_{ave_N} is the required estimated average data traffic for the new packet call N.
- P_i is the estimated average data traffic on the PRCH from packet call i.
- U (pri) are the packet calls with priorities higher than or equal to Pri, where Pri_N is the priority for the requested packet call N.

P_{tol} is the maximum tolerable data traffic on the PRCH.

The above equation is satisfied if average data traffic from packet calls with priority higher than or equal to the priority of the new packet call plus the estimated average data traffic required for the new packet call is less than the maximum tolerable traffic P_{tol}. Thus, a high priority packet call may be allowed to use the PRCH although the total traffic (including all packet calls regardless of priority) exceeds the maximum tolerable traffic P_{tol}. In that case the congestion control function (FIG. **8**) will expel lower priority packet calls so that the total traffic will fall below the maximum tolerable traffic P_{tol}.

The maximum tolerable traffic P_{tol} is associated with a maximum tolerable delay on the PRCH, T_{tol}, according to the relation:

$$P_{tol} = \sum_i P_i + \Delta P$$

$$\Delta P = f(T_{tol} - T)$$

where f is a function having the same sign as its argument and T is the estimate of the average packet delay that is calculated by PRCH traffic supervision function and

$$\sum_i P_i$$

is the sum of the estimated average data traffic for all packet calls on the PRCH.

Because the PRCH controller traffic supervision function continuously monitors T, P_{tol} is continuously updated according to the above equations. P_{tol} will correspond to the traffic level that results in the maximum tolerable delay T_{tol}. In alternatives of the embodiment, the admission control evaluation can be performed using P_{ave_N}, P_i, P_{tol} and ΔP values for the uplink and downlink of the PRCH separately, or, using values for the combined uplink and downlink of the PRCH.

After evaluating the PRCH admission request at Step **710**, the process then moves to Step **712**. At Step **712** the results of Step **710** are checked. If a positive determination in the evaluation was made, the process moves to Step **714** where an admission grant is sent to PRCH manager **402**. If a negative determination was made in the evaluation, the process moves to Step **716** where an admission denied is sent to the PRCH manager **402**. After the PRCH admission control function **416** sends an admission grant or denial at Step **714** or **716**, respectively, the process then moves to Step **818** of FIG. **8A**.

At Step **818** the PRCH congestion control function **418** evaluates congestion on the PRCH. A delay alarm level set by the system operator, T_{con}, and the estimated average packet delay, T, on the PRCH are used to detect a congestion situation, i.e. when it is necessary to expel one or more packet calls from the PRCH in order to regain an acceptable average packet delay on the PRCH.

To evaluate congestion at Step **818** it is determined if $T < T_{con}$. The congestion determination may be made considering uplink and downlink T and T_{con} values in separate determinations, or, using T and T_{con} values for the uplink and downlink combined. Next, at Step **820**, the results of Step **818** are checked. If a positive determination was made at Step **818**, the process returns to the wait state of Step **602** in FIG. **6**. If, however, a negative determination was made at Step **818**, the process moves to Step **822**, where a packet call or, packet calls are selectively chosen for expulsion from the PRCH.

At Step **822** packet calls may be chosen for expulsion by alternative methods. A single packet call may be expelled or, more than one packet call may be expelled from the PRCH at a time.

Referring now to FIG. **8B**, therein are illustrated process steps followed according to an embodiment of the invention when a single packet call is to be expelled at a time by the congestion control function. At Step **826** the lowest priority packet call or packet calls, if more than one exists with the lowest priority, are identified. Next, at Step **828** it is deter-

mined if only one or, if more than one packet call was identified in Step 826. If a determination is made that only one packet call was identified, the process moves to Step 830 and the single identified packet call is selected for expulsion. If, however, a determination is made that more than one packet call having the lowest priority was identified, the process moves to Step 832. At Step 832 one of the identified packet calls is selected for expulsion from the PRCH. The selection of packet call in Step 832 may be done by alternative methods. A packet call may be randomly chosen from the identified packet calls or, one may be chosen based on a comparison using a select parameter associated with each of the packet calls. Depending on the alternative used at Step 818, the selected parameter may be a parameter value for the uplink and downlink separately or, for the combined uplink and downlink.

As an example, one of the following parameters of the lowest priority packet calls, could be selected for comparison:

P_{ave}

P_i

$P_{max(i)}$

$\Delta P_{max} = P_i - P_{max(i)}$.

A packet call could then be selected by choosing the packet call having the largest value of the compared parameter or, the smallest value of the compared parameter, depending on the system operator's desires.

As an alternative method of performing Step 822, more than one packet call may be expelled at a time. Referring now to FIG. 8C, therein are illustrated process steps followed according to an embodiment of the invention when more than one packet call is to be expelled at one time by the congestion control function. At Step 834 a list of packet calls ordered from lowest priority to highest priority is created. Next, at Step 836, an excess traffic value is calculated for the PRCH. The excess traffic value may be calculated as follows:

$$\Delta P = f(T_{tol} - T_{con})$$

where f is a function having the same sign as its argument, T_{tol} equals the maximum tolerable delay of the PRCH and T_{con} equals the threshold as defined above. Depending on the alternative used at Step 818, ΔP may be calculated and checked for the uplink and downlink separately, considering uplink and downlink values of T_{tol} and T_{con}, or, calculated and checked for the uplink and downlink combined using T_{tol} and T_{con} values for the uplink and downlink combined. From Step 836 the process moves to Step 838. At Step 838 packet calls are selected from the list created in Step 834, in order of ascending priority by repeating Steps 840 to 846, until the following equation is satisfied:

$$\sum_{i=1}^{N} P_i \geq \Delta P \text{ where } \sum_{i=1}^{N} P_i$$

is the sum of average data traffic of the selected packet calls and ΔP is the excess data traffic as calculated in Step 836. If more than one packet call exists having the lowest priority, the lowest priority packet calls may be selected for expulsion in random order or, as an alternative, in an order based on a comparison using a select parameter associated with each of the packet calls as was described for Step 832 in FIG. 8B.

After choosing packet calls for expulsion from the PRCH at Step 822, the process then moves to Step 824 and sends a packet call expelled indication for each of the chosen packet calls to the PRCH manager. The process then returns to the wait state of Step 602 in FIG. 6. Upon the next internally generated activation signal indicating a PRCH congestion check should be done or, receiving a packet report, the process will again evaluate congestion on the PRCH and expel additional packet calls, if necessary.

Referring now to FIG. 11, therein is a schematic block diagram illustrating one hardware embodiment of the packet congestion control function 418a of FIG. 4. In the embodiment shown in FIG. 11, the congestion control function comprises a congestion determiner 1102 and a selector 1104. Selector 1004 comprises a packet call selector 1108, sum calculator 1106 for determining if

$$\sum_{i=1}^{N} P_i \geq \Delta P,$$

and an excess data traffic calculator 1110 for determining ΔP. The congestion control function 418a interfaces with PRCH database 1016. The embodiment shown in FIG. 10 is a representative embodiment. It is well known in the art that functions of this type may be implemented in either hardware or software, or in a combination of hardware or software, operating in conjunction with one or more processors.

Referring now to FIG. 9, therein is a flow diagram illustrating process steps followed by the resource manager function according to an embodiment of the invention. The resource manager process is in the wait state of Step 902 when an input is received from the PRCH manager 402. The input may be a PRCH setup request or a PRCH release request. Upon receiving an input, the process moves to Step 904. At Step 904 it is determined if the input is a PRCH setup request. If the input is a PRCH setup request, the process moves to Step 906.

At Step 906 the PRCH setup request is evaluated. The resource manager evaluates the setup request by determining if adequate resources exist within the cell to allow a new PRCH to be set up. From Step 906 the process moves to Step 910. At Step 910 it is determined if the setup request evaluation indicates a new PRCH may be set up. If it is determined that a new PRCH may be set up, the process moves to Step 916 where a PRCH setup grant is sent to PRCH manager 402. Next, at Step 918, the resource manager allocates resources for a new PRCH. From Step 918 the process returns to the wait state of Step 902. If, however, at Step 910 it is determined that the setup request evaluation indicates that a new PRCH may not be set up, the process moves to Step 914 where a PRCH setup denied is sent to PRCH manager 402. From Step 914 the process returns to the wait state of Step 902.

If the input is determined not to be a PRCH setup request at Step 904, it is a PRCH release request. In this case the process moves from Step 904 to Step 912. At Step 912 the PRCH release request is evaluated. The resource manager evaluates the PRCH release request by determining whether it is acceptable to release the PRCH from an overall system point of view. For example, the traffic load on PRCHs of surrounding cells could be taken into account. From Step 912 the process moves to Step 920. At Step 920 it is determined if the PRCH release request evaluation indicates a PRCH may be released. If it is determined that the PRCH may be released, the process moves to Step 922 where a PRCH release grant is sent to PRCH manager 402. Next, at Step 926, the resource manager releases the PRCH. From Step 926 the process returns to the wait state of Step 902. If, however, at Step 920, it is determined the PRCH release

request evaluation indicates that the PRCH may not be released, the process moves to Step 924 where a PRCH release denied is sent to PRCH manager 402. From Step 924 the process returns to the wait state at Step 902.

As can be seen from the above description, the method and system of the invention can be used by a system operator to manage packet traffic for prioritized users on one or more PRCHs of a cellular telecommunications system. The system operator can set a maximum average time delay for the PRCH. The users can be prioritized according to a level of service subscribed to or a priority could be assigned automatically or chosen by the user depending on the type of call being made. A higher priority level may incur a higher charge rate for using the system. Paying the higher rate allows the user to be prioritized before other uses having lower priorities in congestion situations and when trying to access the system. By making packet traffic management decisions based on the estimated data traffic required by the packet call and the priority of the packet call, a system operator can be assured that PRCH users are not subject to unacceptable PRCH delays.

It is believed that the operation and construction of the present invention will be apparent from the foregoing description and, while the invention shown and described herein has been described as a particular embodiment, changes and modifications may be made therein without departing from the spirit and scope of the invention as defined in the following claims.

What is claimed is:

1. In a telecommunication system comprising at least one packet radio channel and a plurality of transceiving stations, each of said transceiving stations capable of transmitting packet calls comprising a plurality of data packets on at least one packet radio channel, a method of supervising traffic on the at least one radio channel, said method comprising the steps of:

a) receiving periodic packet reports associated with at least one packet call on the packet radio channel, said packet report including information on a number of frames included in the packet call, a packet identifier for identifying the packet call associated with the packet report and a time stamp indicating when the packet call was placed in a transmission buffer;

b) calculating the size of the packet call in time in response to the information on the number of frames;

c) calculating the difference between the time the packet report was received and the time the packet call was placed in the transmission buffer;

d) calculating the time lapse since a previous packet report having the same packet identifier was received; and

e) utilizing the calculations from steps b) –d), to calculate an estimated average data traffic for the packet call on the at least one packet radio channel.

2. The method of claim 1, further including the step of determining an estimate of average data traffic for all packet calls on the at least one packet radio channel from the information calculated in steps b)–e) upon receipt of each packet report.

3. The method of claim 2 in which said packet calls are transmitted on an uplink of said packet radio channel and said step of calculating average data traffic comprises the step of calculating average data traffic indicative of traffic on said uplink.

4. The method of claim 2 in which said packet calls are transmitted on a downlink of said packet radio channel and said step of calculating average data traffic comprises the step of calculating average data traffic indicative of traffic on said downlink.

5. The method of claim 2 in which said packet calls are transmitted on the downlink and uplink of said packet radio channel and said step of calculating average data traffic comprises the step of calculating average data traffic indicative of combined traffic on said downlink and uplink.

6. The method of claim 1, further including the step of calculating an estimate of the average packet delay for the packet call on the at least one packet radio channel in response to the calculations of steps b)–d).

7. The method of claim 6 in which said packet calls are transmitted on an uplink of said packet radio channel and said step of calculating average packet delay further comprises the step of calculating average packet delay indicative of delay on said uplink.

8. The method of claim 6 in which said packet calls are transmitted on a downlink of said packet radio channel and said step of calculating average packet delay comprises the step of calculating average packet delay indicative of delay on said downlink.

9. The method of claim 6 in which said packet calls are transmitted on the downlink and uplink of said packet radio channel and said step of calculating average packet delay comprises the step of calculating average delay indicative of combined delay on said downlink and uplink.

10. The method of claim 1, further including the step of calculating an estimate of the average packet delay on the at least one packet radio channel in response to the calculations of steps b)–d) upon receipt of each packet report.

11. The method of claim 1, further including the step of utilizing the value of the estimated average data traffic for all packet calls for controlling admission control and congestion control processes to the at least one packet radio channel in accordance with each newly received packet report.

12. In a telecommunication system comprising at least one packet radio channel and a plurality of transceiving stations, each of said transceiving stations capable of transmitting and receiving packet calls on the at least one packet radio channel, a method of supervising traffic on the at least one radio channel, said method comprising the steps of:

a) receiving periodic packet reports associated with a packet call on the packet radio channel, said packet report including information on a number of frames a for the associated packet call, a packet identifier for identifying the packet call associated with the packet report and a time stamp indicating when the associated packet call was placed in a transmission buffer;

b) calculating the size of the packet call in time in response to the information on the number of frames;

c) calculating the difference between the time the packet report was received and the time the data packet was placed in the transmission buffer;

d) calculating the time lapse since a previous packet report having the same packet identifier was received; and

e) utilizing the calculations from steps b)–d), to calculate an estimated average traffic delay for each individual packet call on the at least one packet radio channel upon receipt of each packet report.

13. In a telecommunication system comprising at least one packet radio channel and a plurality of transceiving stations, each of said transceiving stations capable of transmitting and receiving packet calls comprising a plurality of data packets on the at least one packet radio channel, a

method of supervising traffic on the at least one radio channel, said method comprising the steps of:

a) receiving periodic packet reports associated with packet calls on the packet radio channel, said packet report including information on a number of frames included in the associated packet call, a packet identifier for identifying the packet call associated with the packet report and a time stamp indicating when the associated packet call was placed in a transmission buffer;

b) calculating the size of the packet call in time in response to the number of frames;

c) calculating the difference between the time the packet report was received and the time the packet call was placed in the transmission buffer;

d) calculating the time lapse since a previous packet report having the same packet identifier was received; and

e) utilizing the calculations from steps b)–d), to calculate an estimated average data traffic delay for each individual packet call on the at least one packet radio channel upon receipt of each packet report.

14. An apparatus for controlling traffic on a packet radio channel, said apparatus comprising:

means for receiving periodic packet reports on a packet call from a network protocol stack, the packet report including a packet identifier, a number of frames comprising the packet call and a time stamp indicating when the packet call was placed in a transmission buffer;

an admission control function for controlling admission of packet calls to the packet radio channel;

a congestion control function for controlling expulsion of packet calls from the packet radio channel; and

a packet radio channel traffic supervisor function responsive to the number of frames and the time stamp in each received packet report for updating system traffic status and controlling the admission control function and the congestion control function in accordance with the updated system traffic status.

15. The apparatus of claim 14 wherein the packet radio channel supervisor function further comprises:

first means for calculating the size of the packet call in time from the number of frames in the packet report;

second means for calculating a packet delay for the data packet call from the time stamp, the packet delay representing a difference between a time the packet report was received and a time the packet was placed in the transmission buffer; and

third means for calculating an elapsed time since a packet report with same packet identifier was received.

16. The apparatus of claim 15, further including means responsive to the size of the packet call, the packet delay, and the elapsed time between packet reports for determining an average data traffic for the packet call.

17. The apparatus of claim 15, further including means responsive to the size of the packet call, the packet delay, and the elapsed time between packet reports for determining an average data traffic for the packet radio channel.

18. The apparatus of claim 15, further including means responsive to the size of the packet call, the packet delay, and the elapsed time between packet reports for determining an average packet delay caused by the packet call.

19. The apparatus of claim 15, further including means responsive to the size of the packet call, the packet delay, and the elapsed time between packet reports for determining an average packet delay for the packet radio channel upon receipt of each packet report.

* * * * *

US005517495A

United States Patent [19]

Lund et al.

[11] Patent Number: 5,517,495

[45] Date of Patent: May 14, 1996

[54] **FAIR PRIORITIZED SCHEDULING IN AN INPUT-BUFFERED SWITCH**

[75] Inventors: **Carsten Lund**, New Providence, N.J.; **Steven Phillips**, New York, N.Y.; **Nicholas F. Reingold**, Madison, N.J.

[73] Assignee: **AT&T Corp.**, Murray Hill, N.J.

[21] Appl. No.: **350,347**

[22] Filed: **Dec. 6, 1994**

[51] **Int. Cl.^{6}** **H04L 12/56**

[52] **U.S. Cl.** **370/60**; 370/85.6

[58] **Field of Search** 370/60, 60.1, 85.6, 370/94.1, 94.2, 61

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,166,926	11/1992	Cisneros et al.	370/94.1
5,255,265	10/1993	Eng et al.	370/60
5,287,182	2/1994	Haskell et al.	348/500
5,325,356	6/1994	Lyles	370/60

OTHER PUBLICATIONS

T. E. Anderson et al., "High-Speed Scheduling for Local-Area Networks", ACM Trans. on Computer Systems, vol. 11, No. 4, Nov. 1993, pp. 319–352.

M. J. Karol et al., "Input Versus Output Queueing on a Space-Division Packet Switch", IEEE Trans. on Comm., vol. COM-35, No. 12, 1987, pp. 1347–1356.

N. McKeown et al., "Scheduling Cells in an Input-Queued Switch", Electronics Lett., Dec. 1993, pp. 1–4.

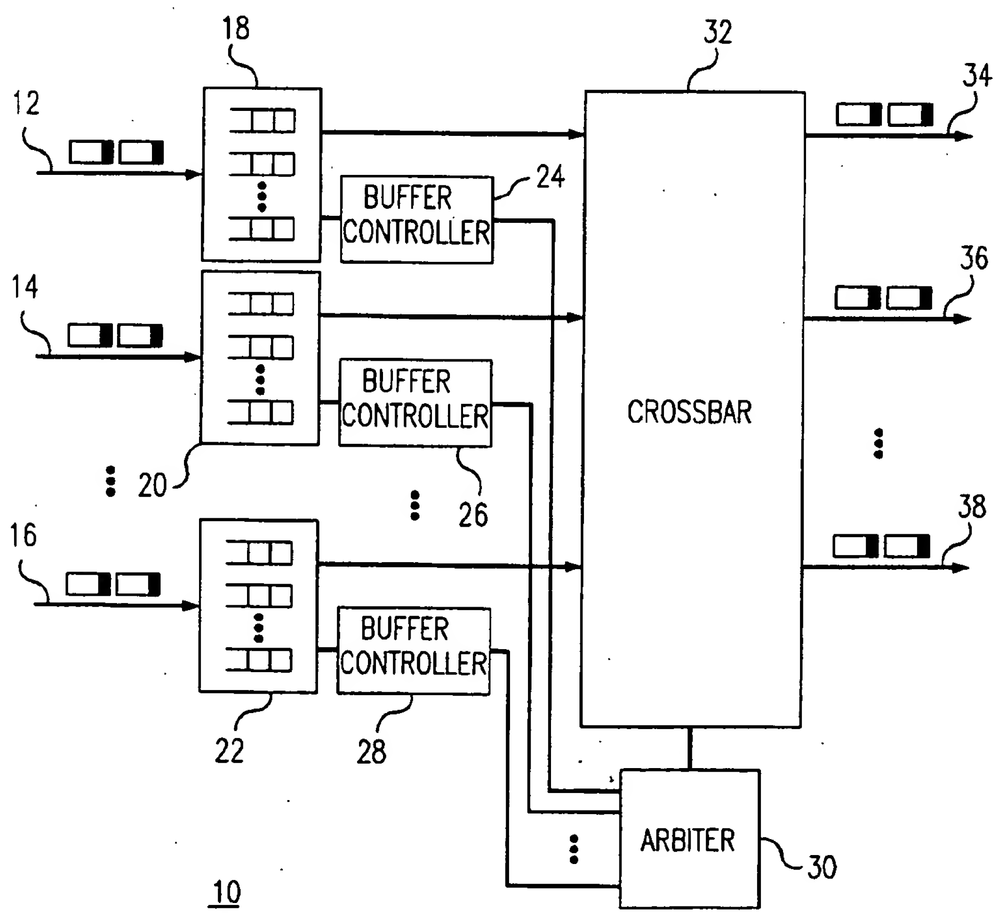
H. Obara et al., "An Efficient Contention Resolution Algorithm for Input Queueing ATM Cross-Connect Switches", Int'l. J. of Digital and Analog Cabled Systems, vol. 2, 1989, pp. 261–267.

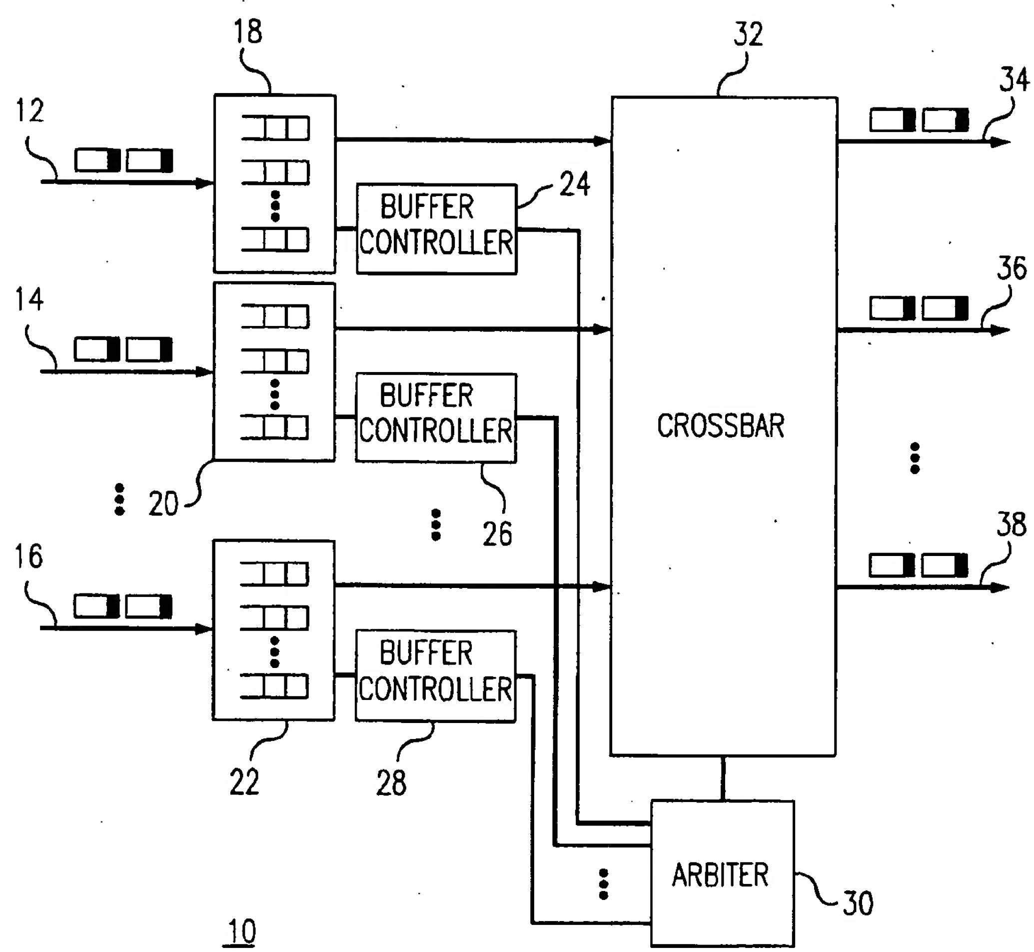
Primary Examiner—Melvin Marcelo

[57] **ABSTRACT**

A Fair Arbitrated Round Robin (FARR) method is disclosed for scheduling the crossbar of an input-buffered asynchronous transfer mode (ATM) switch using an arbiter. Per-virtual-circuit queuing of ATM cells is performed.

23 Claims, 11 Drawing Sheets





18
32
34
12
24
BUFFER CONTROLLER
36
14
BUFFER CONTROLLER
CROSSBAR
20
26
38
16
BUFFER CONTROLLER
22
28
ARBITER
30
10

FIG. 1

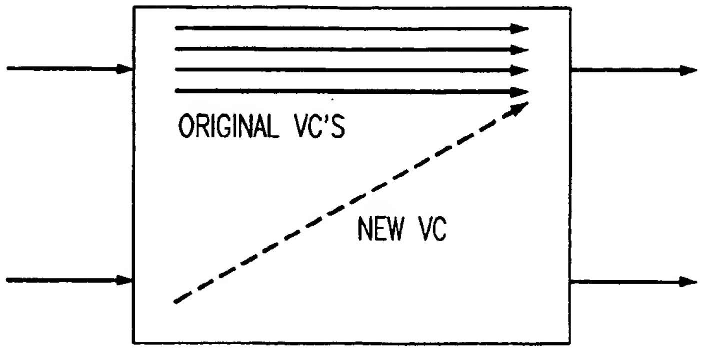


FIG. 2

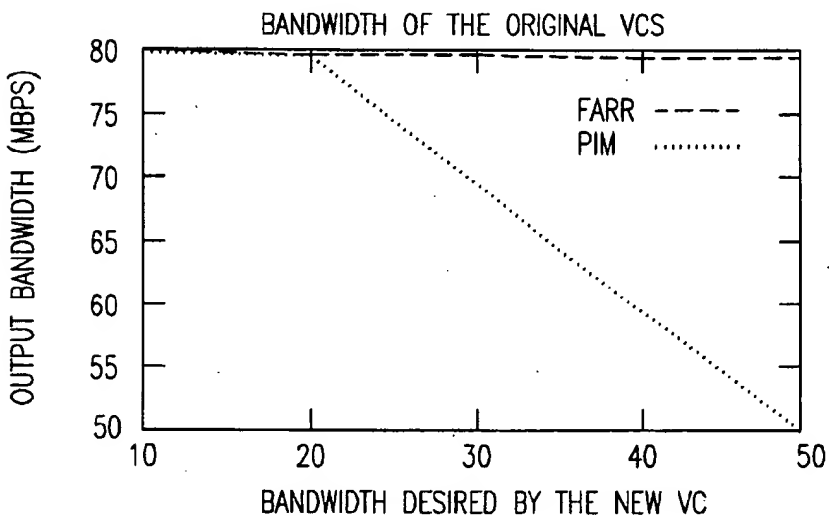


FIG. 3

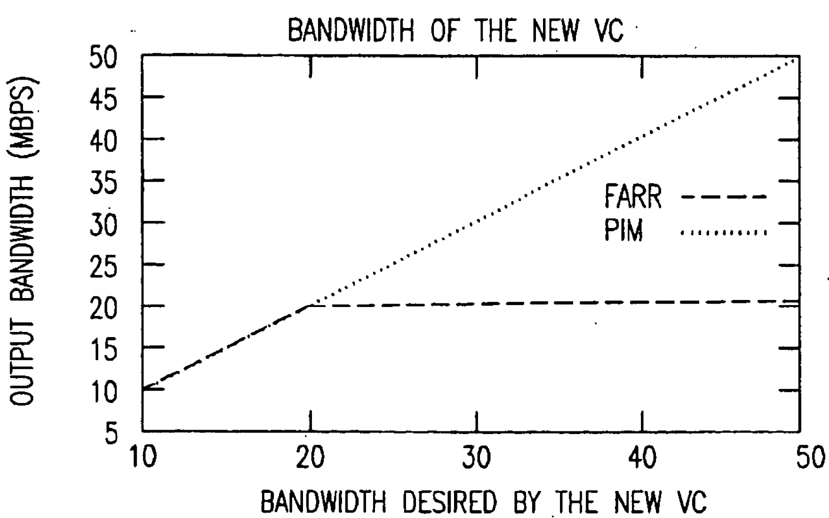


FIG. 4

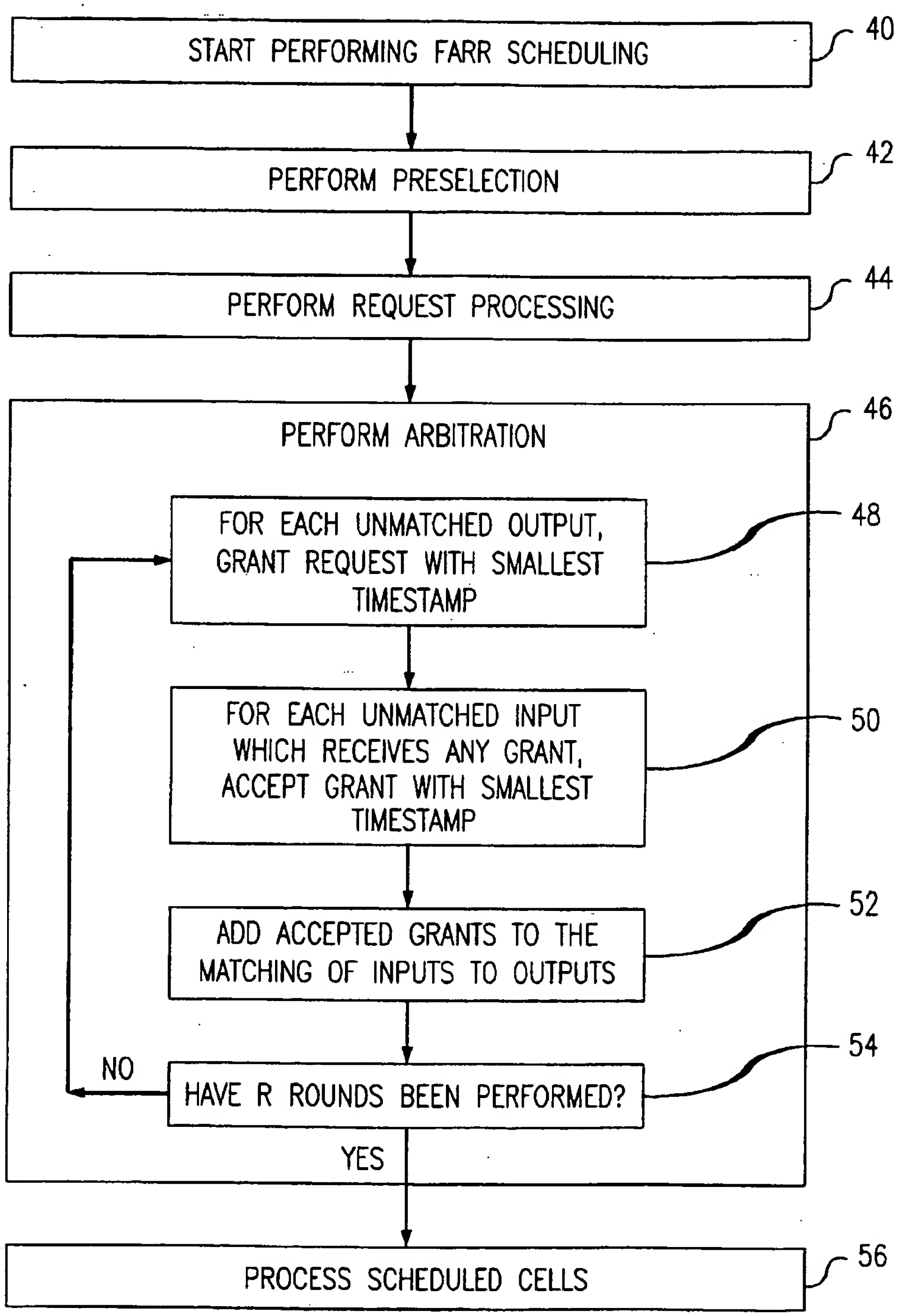

START PERFORMING FARR SCHEDULING
40
PERFORM PRESELECTION
42
PERFORM REQUEST PROCESSING
44
PERFORM ARBITRATION
46
FOR EACH UNMATCHED OUTPUT, GRANT REQUEST WITH SMALLEST TIMESTAMP
48
FOR EACH UNMATCHED INPUT WHICH RECEIVES ANY GRANT, ACCEPT GRANT WITH SMALLEST TIMESTAMP
50
ADD ACCEPTED GRANTS TO THE MATCHING OF INPUTS TO OUTPUTS
52
NO
HAVE R ROUNDS BEEN PERFORMED?
54
YES
PROCESS SCHEDULED CELLS
56

FIG. 5

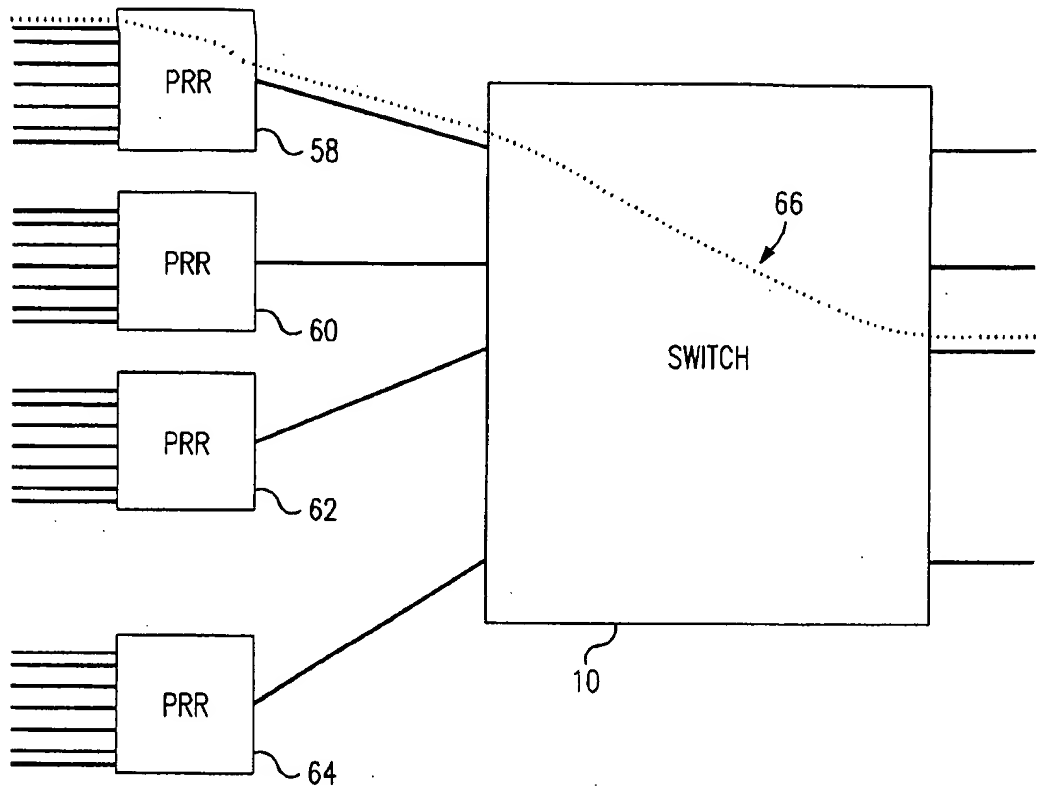

PRR
PRR
PRR
PRR
58
60
62
64
66
SWITCH
10

FIG. 6

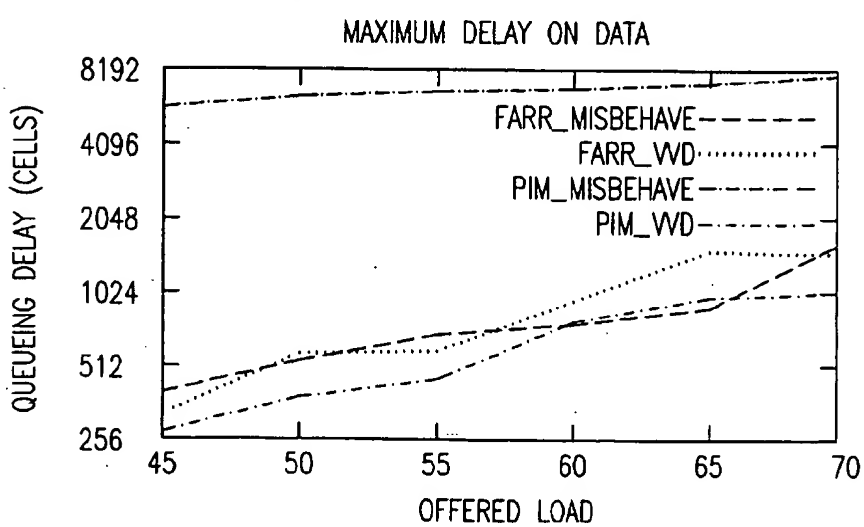

MAXIMUM DELAY ON DATA
QUEUEING DELAY (CELLS)
8192
4096
2048
1024
512
256
45
50
55
60
65
70
OFFERED LOAD
FARR_MISBEHAVE
FARR_WD
PIM_MISBEHAVE
PIM_WD

FIG. 7

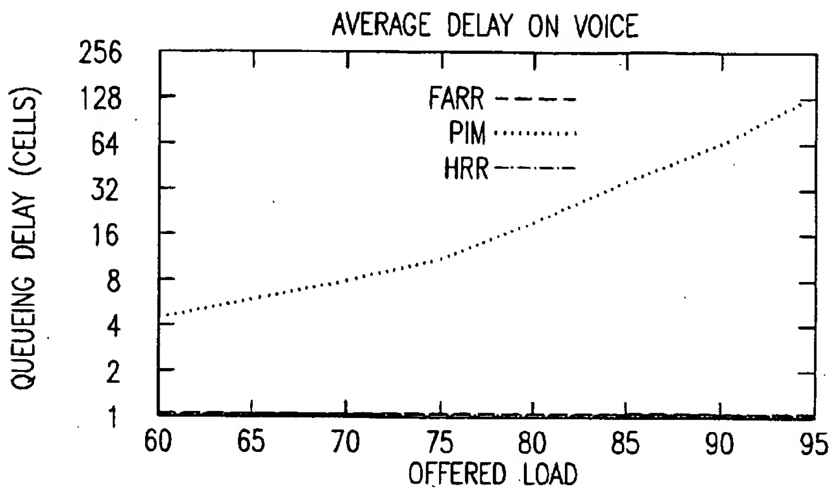


FIG. 8

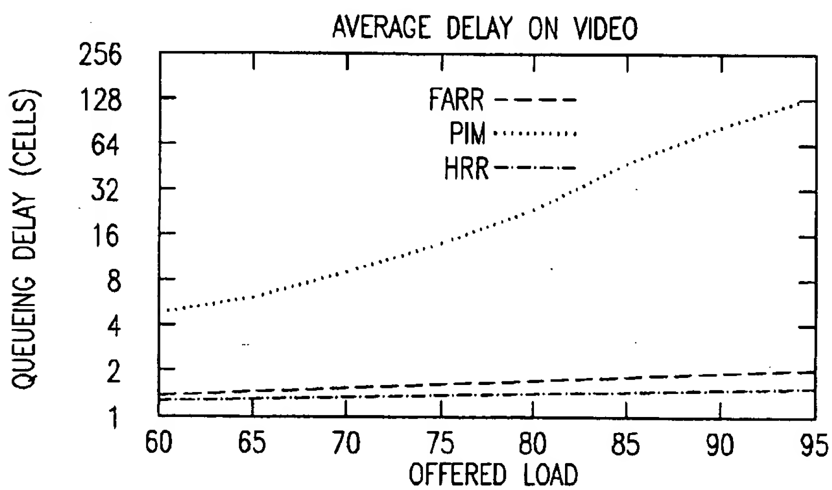


FIG. 9

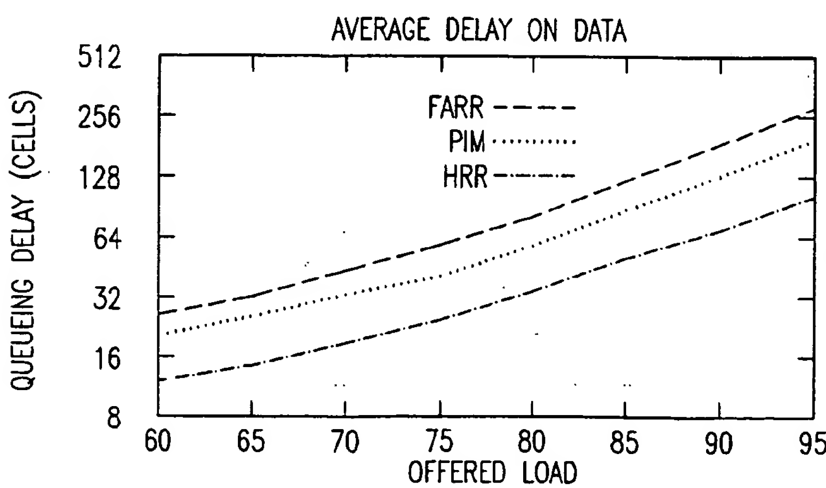


FIG. 10

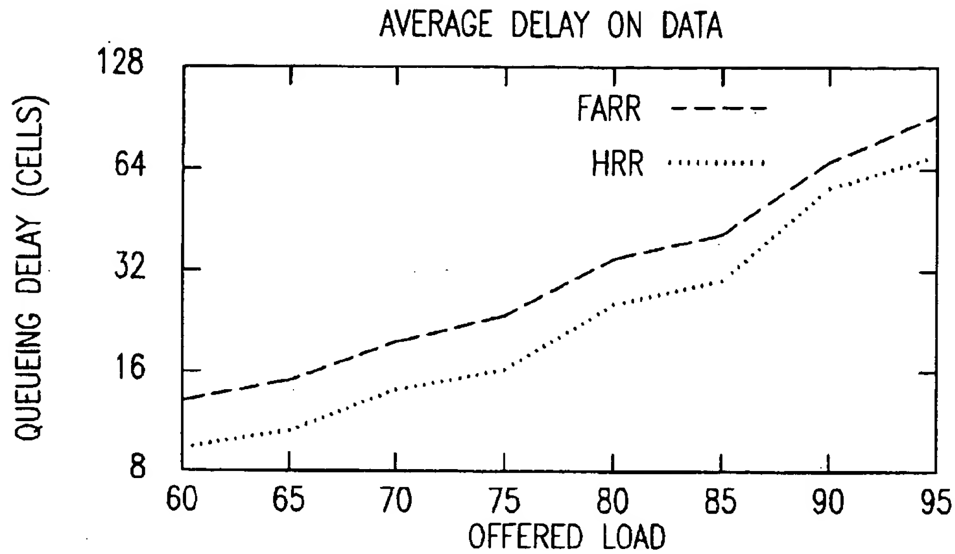

AVERAGE DELAY ON DATA
QUEUEING DELAY (CELLS)
128
64
32
16
8
FARR
HRR
60
65
70
75
80
85
90
95
OFFERED LOAD

FIG. 11

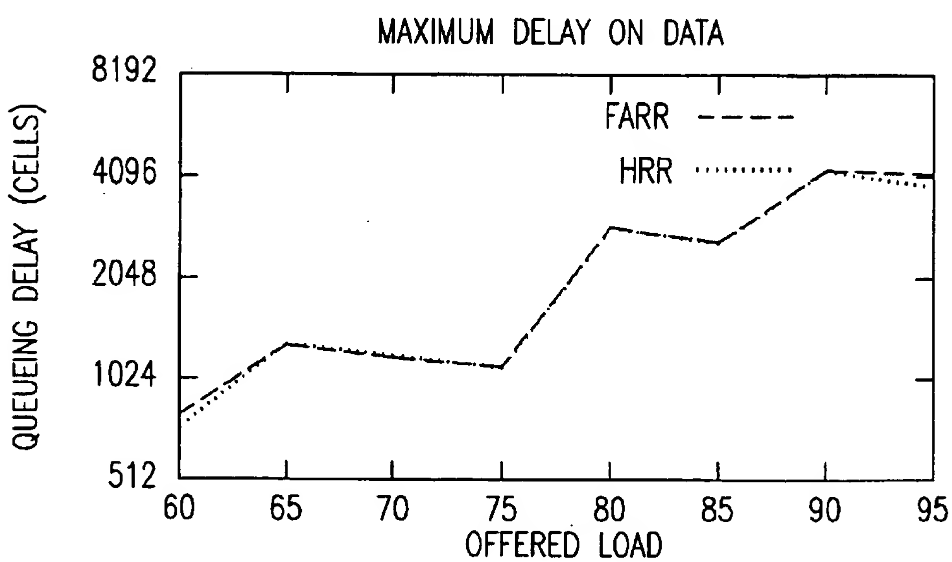

MAXIMUM DELAY ON DATA
QUEUEING DELAY (CELLS)
8192
4096
2048
1024
512
FARR
HRR
60
65
70
75
80
85
90
95
OFFERED LOAD

FIG. 12

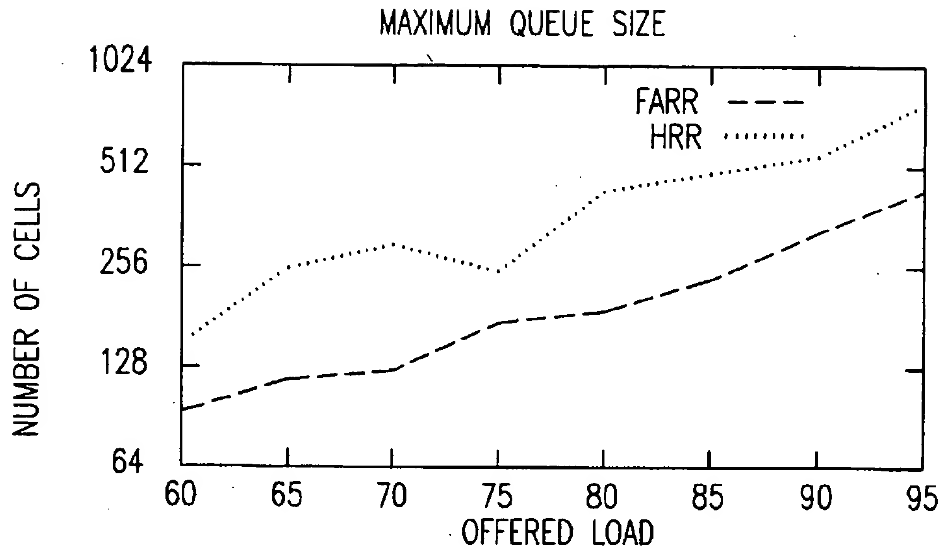

MAXIMUM QUEUE SIZE
1024
512
256
128
64
NUMBER OF CELLS
FARR
HRR
60
65
70
75
80
85
90
95
OFFERED LOAD

FIG. 13

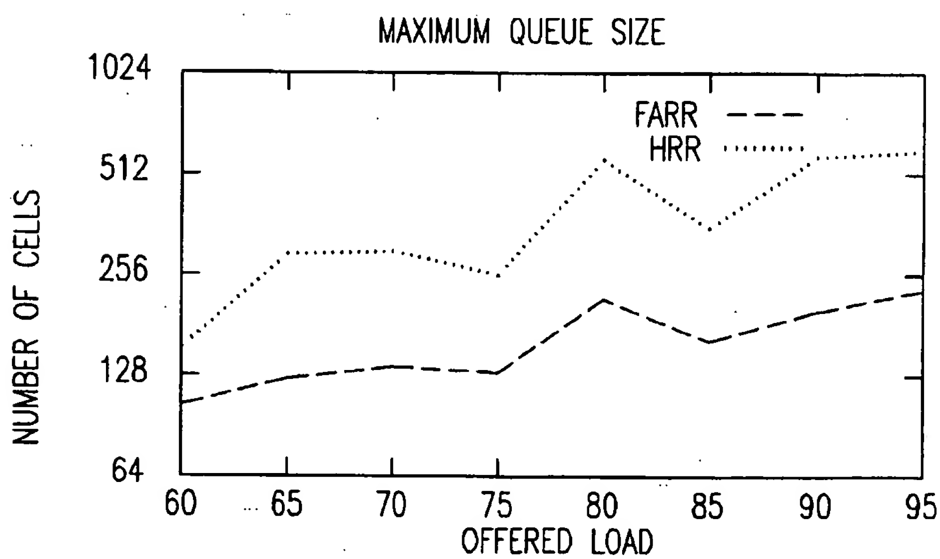

MAXIMUM QUEUE SIZE
1024
512
256
128
64
NUMBER OF CELLS
FARR
HRR
60
65
70
75
80
85
90
95
OFFERED LOAD

FIG. 14

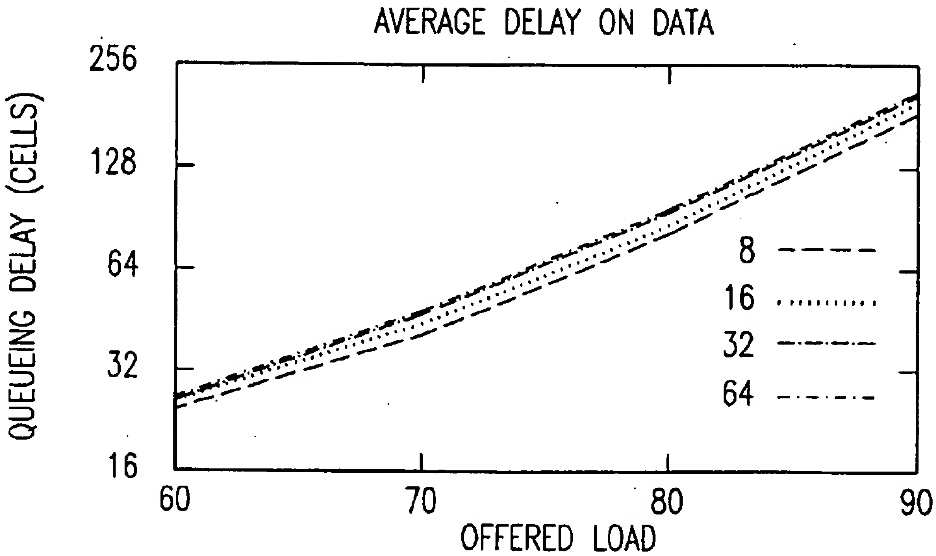

AVERAGE DELAY ON DATA
QUEUEING DELAY (CELLS)
256
128
64
32
16
60
70
80
90
OFFERED LOAD
8
16
32
64

FIG. 15

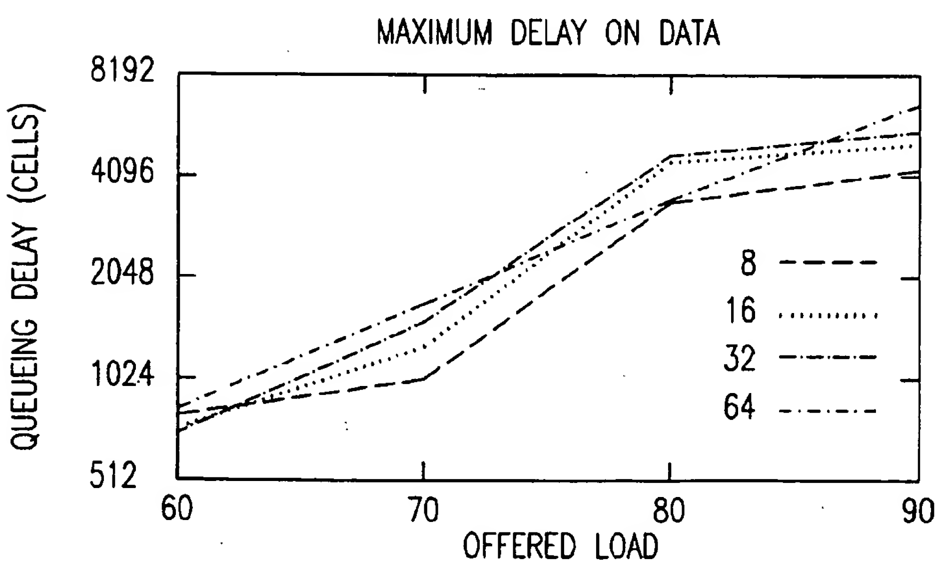

MAXIMUM DELAY ON DATA
QUEUEING DELAY (CELLS)
8192
4096
2048
1024
512
60
70
80
90
OFFERED LOAD
8
16
32
64

FIG. 16

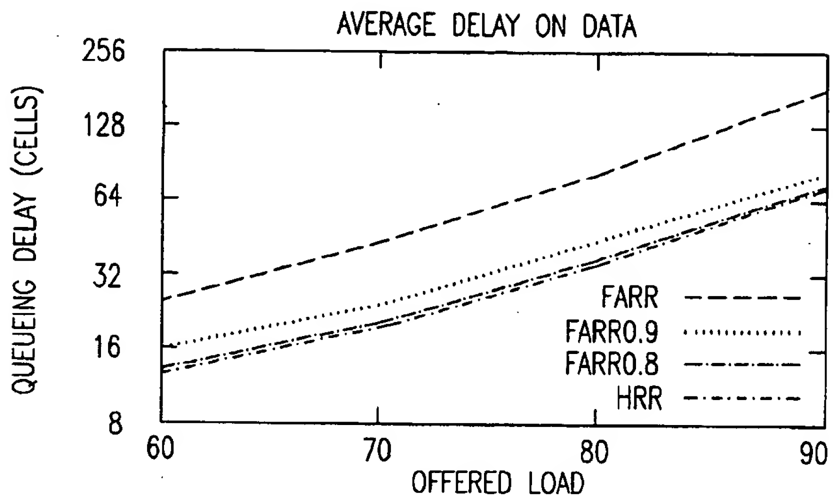


FIG. 17

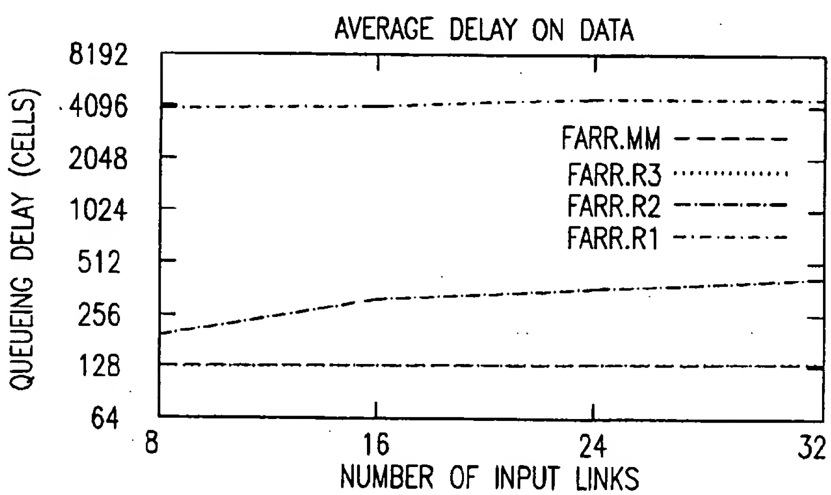


FIG. 18

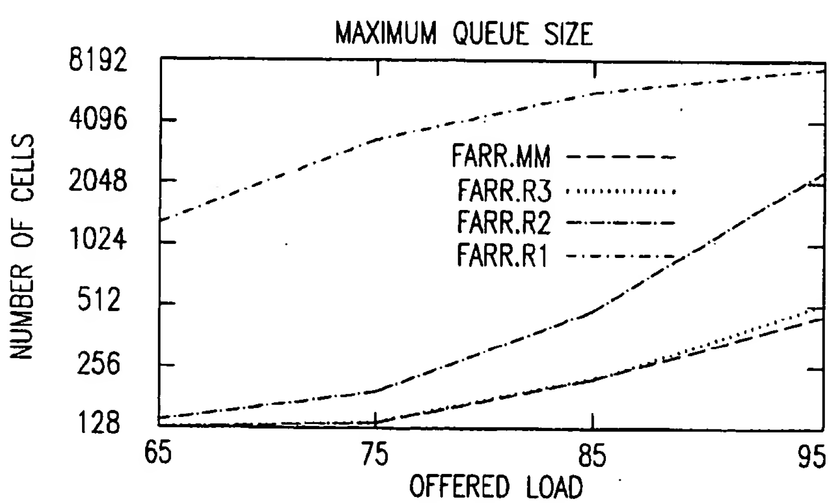


FIG. 19

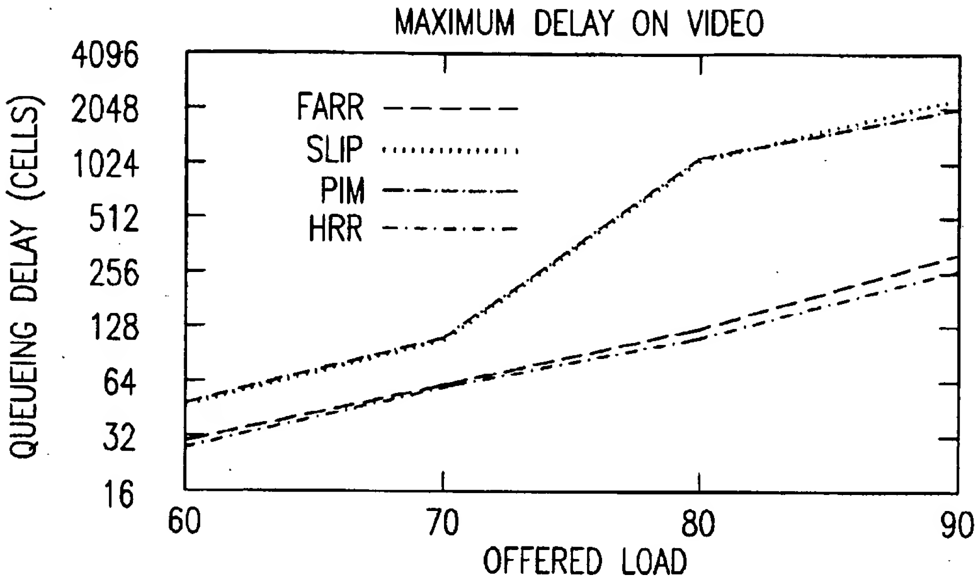

MAXIMUM DELAY ON VIDEO
QUEUEING DELAY (CELLS)
4096
2048
1024
512
256
128
64
32
16
FARR
SLIP
PIM
HRR
60
70
80
90
OFFERED LOAD

FIG. 20

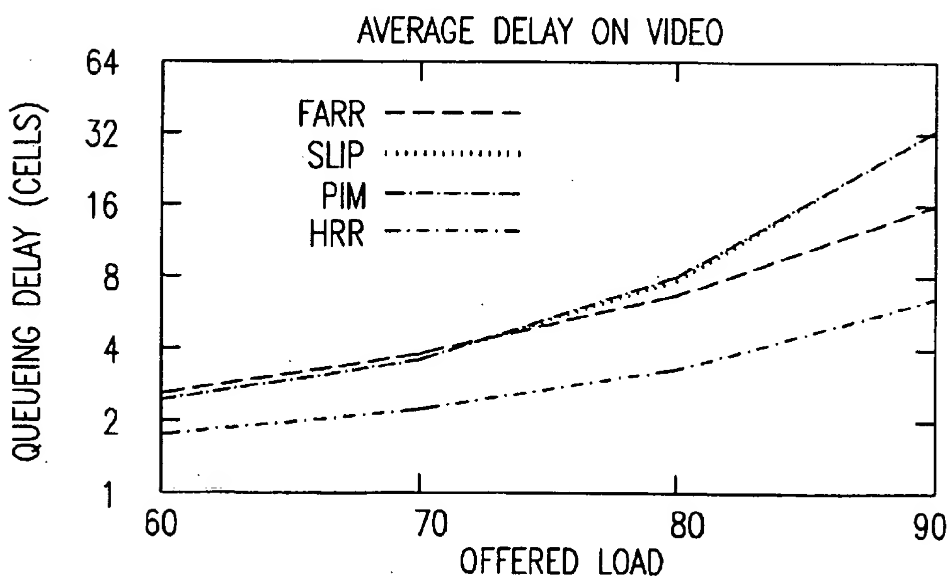

AVERAGE DELAY ON VIDEO
QUEUEING DELAY (CELLS)
64
32
16
8
4
2
1
FARR
SLIP
PIM
HRR
60
70
80
90
OFFERED LOAD

FIG. 21

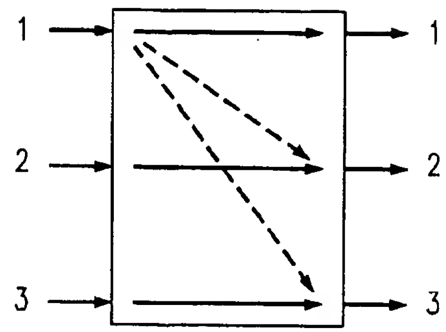


FIG. 22

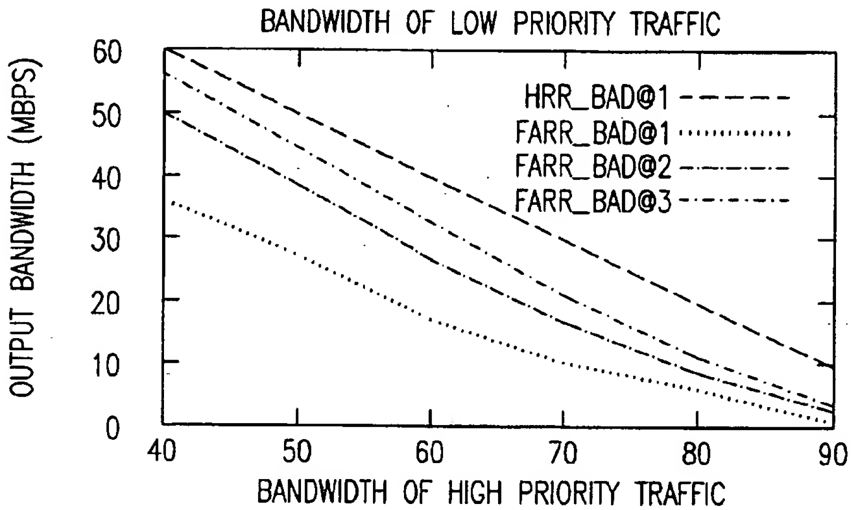


FIG. 23

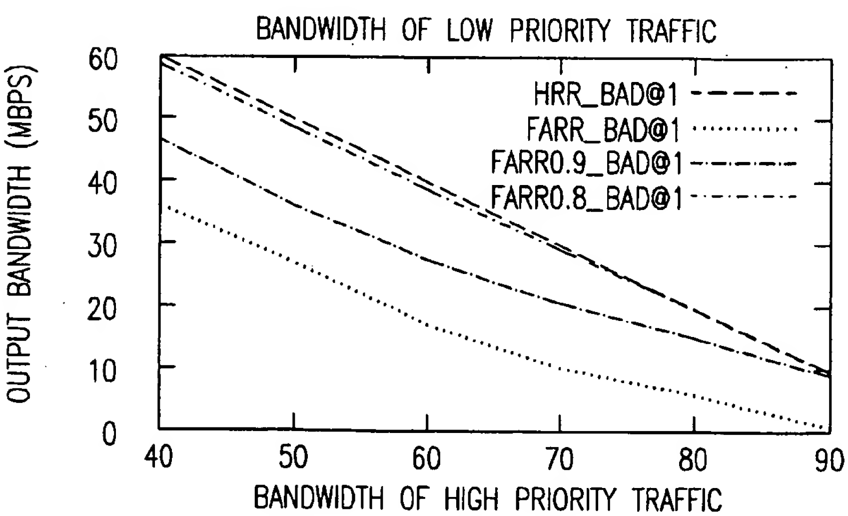


FIG. 24

FAIR PRIORITIZED SCHEDULING IN AN INPUT-BUFFERED SWITCH

BACKGROUND OF THE INVENTION

1. Field of the Invention

This disclosure relates to fixed size packet switches, such as asynchronous transfer mode (ATM) switches, and in particular to an apparatus and method for scheduling cell inputs through ATM switches.

2. Description of the Related Art

Advances in high-speed cell-based communications have been prompted by a number of factors, including increased traffic on the "Information Superhighway", the high bandwidth requirements of video-on-demand, the convergence of technology trends enabling high-speed communications, such as laser-optic technology, and the progress of ATM communications which makes possible the unification of the transmission of voice, video, and data in a common technology.

ATM offers uniformity of network protocols with respect to both cell content (voice, data, etc.) and network scale. Standardized cell formats in ATM support voice traffic in large commercial telephone networks and simplify the design of switches for smaller networks.

Using round robin scheduling, an output-buffered switch may run at multi-gigabit speeds but is prohibitively expensive to implement. Generally, fair prioritized round robin (PRR) scheduling provides fair resolution when there is conflict for resources. Such PRR scheduling has been discussed in, for example, E. L. Hahne, "Round Robin Scheduling for Fair Flow Control in Data Communication Networks", LABORATORY FOR INFORMATION AND DECISION SYSTEMS, Mass. Inst. Tech., Cambridge, Mass. 02139.

For input buffered switches, methods such as bandwidth reservation and static scheduling have been implemented to provide cell scheduling. Another scheduling method, Iterative Round Robin Matching with Slip, is discussed in McKeown et al., "Scheduling Cells in an Input-Queued Switch," ELECTRONICS LETTERS, December 1993. Another scheduling method is Parallel Iterative Matching (PIM) discussed in T. E. Anderson et al., "High-Speed Switch Scheduling for Local-Area Networks," ACM TRANS. ON COMPUTER SYSTEMS, 11(4):319–352, 1993. PIM switch scheduling methods allocating bandwidth fairly among the links may be very unfair in allocating bandwidth between virtual circuits.

SUMMARY

An input-buffered ATM switch is disclosed with novel switch scheduling method which is suitable for fast low cost local to wide area networks. The ATM switch includes a plurality of input buffers for receiving the arriving cells, each input buffer associated with a respective input port; a cell switching fabric for processing the received cells from the input buffers to the output ports; and a scheduling control circuit for controlling the processing of the received cells through the cell switching fabric using a fair arbitration round robin (FARR) program. Each input buffer includes a service list associated with each priority level for each output port; and a plurality of cell queues, one per virtual circuit that enters the switch at that input and having an associated timestamp; and the scheduling control circuit uses the timestamps to generate a matching of input buffers to output ports to control the processing of the received cells.

A method is also disclosed for switching arriving cells using FARR in an input-buffered ATM switch from input ports having associated input buffers to output ports. The method includes the steps of receiving cells in the input buffers; in each input buffer, pre-selecting a virtual circuit for each output port; performing request processing to send the timestamp of each pre-selected virtual circuit to a scheduling control circuit; generating a matching of the input buffers to the output ports from the scheduling control circuit using the timestamps; and for each pair in the matching, sending a cell of the virtual circuit corresponding to that pair through a cell switching fabric and setting the timestamp of each virtual circuit corresponding to a pair in the matching.

The step of matching includes the step of using an arbiter unit operating a FARR program. The step of generating a matching includes the steps of, for each unmatched output port, granting the request for it having minimum timestamp; for each unmatched input buffer, accepting the granted request with minimum timestamp; and incorporating the accepted grants in the matching. In addition, the method includes the steps of granting requests, accepting grants, and incorporating the accepted grants are performed for a predetermined number of rounds.

The switch scheduling by FARR respects cell priorities, enables voice and video cells to be transmitted with very small delay, and allows the remaining bandwidth to be used by data cells with less rigorous timing constraints. Contention between virtual circuits within the same priority level is resolved fairly. Each input link has a random access cell buffer and a buffer manager for header translation, addressing, and management of service lists.

The disclosed FARR method performs input-buffered switch scheduling with improved performance over other input-buffered switch scheduling methods such as SLIP, PIM, etc.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the disclosed input-buffered switch and fair arbitrated round robin scheduling method will become more readily apparent and may be better understood by referring to the following detailed description of an illustrative embodiment of the present invention, taken in conjunction with the accompanying drawings, where:

FIG. **1** illustrates an input-buffered switch;

FIGS. **2–4** are graphical illustrations of fairness of per-virtual-circuits;

FIG. **5** is a flowchart of the disclosed method of FARR scheduling;

FIG. **6** illustrates a workload model; and

FIGS. **7–24** illustrate workload evaluation results for the FARR scheduling method and comparable scheduling methods.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now in specific detail to the drawings, with like reference numerals identifying similar or identical elements, as shown in FIG. **1**, the present disclosure describes an input-buffered ATM switch **10** implementing a Fair Arbitrated Round Robin (FARR) cell scheduling method for switching ATM cells. Each input link **12–16** is respectively

associated with one of the random access cell input buffers **18–22** and one of the buffer controllers **24–28** for header translation, addressing, and management of service lists.

An exemplary cell input buffer **18** is the DATAPATH chip available from AT&T Corp., which has a bandwidth of 1.2 Gbits/second and multiplexes/demultiplexes at rates around 155 Mbits/second. The arbiter chip **30** implementing FARR can be constructed, for example, using cell array technology, such as AT&T's HL400C CMOS Standard Cell family. A set of commercially available crossbar chips form the crossbar **32** as the cell switch fabric connected to outputs **34–38**. In the exemplary embodiment, an 8×8 switch is provided with link speeds of 1.2 Gbits/second. It is understood that larger capacity switches such as a 32×32 switch may be implemented in accordance with the present disclosure. Faster buffers permit the switch **10** to be scaled to a link speed of 2.4 Gbits/second and beyond.

In alternative embodiments, the addition of a DATAPATH cell buffer to each output allows the 8×8 switch to be used in a 64×64 configuration, with link speeds of 155 Mbits/second. Such a switch is applicable to many high-speed local area network applications. For large networks, switches may be combined in an arbitrary-topology network, with gigabit/second links between switches. Such a network offers much greater aggregate bandwidth than linear networks such as Ethernet, such as discussed in R. Metcalfe et al., "Ethernet: Distributed Packet Switching for Local Computer Networks", COMM. OF THE ACM, 7:395–404, July 1976.

The disclosed FARR scheduling method is implemented in an input-buffered switch to perform in a manner similar to PRR, where available bandwidth is shared fairly between different virtual circuits (VC) at the same priority level. Thus fairness is inherent in the disclosed scheduling system and method.

INPUT BUFFERING AND MATCHING

For some input-queued switches, head-of-line blocking reduces the performance of the cell processing. For example, if FIFO queues are used at the inputs, then the queues saturate at a utilization value approaching 2–2½= 0.586 for large switches.

Output-queued switches using a switch memory fabric and output buffers operate at N times a link speed in an N×N switch, causing such output-queued switches to become prohibitively expensive at high link speeds. An input-queued switch is less expensive to operate since the switch fabric, such as a crossbar or Batcher-banyan network, is required to operate at the link speed.

As shown in FIG. **1** to implement input-queued switching, random access cell buffers **18–22** are provided at the inputs **12–16** for performing buffering referred to as input-buffering, as discussed in T. E. Anderson et al., cited above.

As disclosed herein, the input-buffer system and method provides fast and effective scheduling of cell processing at the crossbar **32** using a centralized arbiter **30**. In the exemplary embodiment, the arbiter **30** processes fixed-size ATM cells, se the crossbar **32** may operate synchronously. For each cell time, the arbiter **30** selects a set of cells to enter the crossbar **32**. Each cell resides in a different input buffer, and each cell is destined for a different output, so the arbiter **30** repeatedly solves a bipartite matching problem. The arbiter **30** finds a large match in each step, which provides good fairness properties. In the context of ATM, it is desirable to have per-VC fairness, where available bandwidth is shared among contending virtual circuits and where all virtual circuits with the same bandwidth and priority level experience similar maximum cell delays. In addition, cells of high priority have precedence over lower priority cells.

In the disclosed ATM switch, per-virtual circuit fairness is implemented using the FARR method disclosed herein. For example, referring to FIGS. **2–4**, for a 2×2 switch having 100 Mbit links and **4** virtual circuits running from INPUT1 to OUTPUT1, each using 20 Mbits/second of bandwidth, the introduction of a new virtual circuit from INPUT2 to OUTPUT1 shown in FIG. **2**, with a desired bandwidth varying between 10 Mbits/second and 50 Mbit/second has an effect on the bandwidth of the original circuits as shown in FIGS. **3–4**. For the exemplary switch using FARR having per-VC fairness, the bandwidth to the new circuit is limited to 20 Mbits/second, so bandwidth allocation is fair.

THE FARR SCHEDULING METHOD

In the disclosed scheduling method, a matching of inputs to outputs is determined in each cell time, describing the set of cells which is to traverse the crossbar. For the disclosed input-buffered switch, the disclosed FARR method implements the performance of PRR for output-buffered switches such as the PRR implementations cited above.

In the exemplary embodiment, each of input buffers **18–22** maintains a service list for each priority level for each output **34–38**, and each VC in a service list has a timestamp which indicates the time when that VC last sent a cell across the crossbar **32**. The extended timestamp of a VC is its timestamp prepended with its priority. The priority of a virtual circuit is fixed and represents the traffic class. Such priority values may be set as 0 for voice traffic, **1** for video traffic, and **2** for data traffic.

As shown in FIG. **5**, the FARR method for the disclosed input-buffered switch **10** includes the steps of starting the FARR scheduling method in step **40**, performing preselection at the input buffers in step **42**, performing request processing in step **44**, and using the arbiter **30** to perform arbitration in step **46** as follows:

1. in performing preselection in step **42**, each input preselects the VC at the head of the highest priority non-empty service list for each output for which the output has buffered cells;
2. in the processing of requests in step **44**, each input sends the extended timestamp of each preselected VC to the arbiter **30**; and
3. in the performing arbitration in step **46**, the following steps **48–52** are repeated R times until R rounds have been completed in step **54**:
   (a) for each unmatched output, if it has any request from unmatched inputs, FARR method grants the request with smallest extended timestamp in step **48**;
   (b) for each unmatched input, if it receives any grants, the FARR method accepts the grant with smallest extended timestamp in step **50**; and
   (c) any accepted grants are added to the matching in step **52**.

Upon completing the arbitration in step **46** to obtain a matching of inputs to outputs, the FARR method processes the matching in step **56** by sending the oldest cell of each virtual circuit in the matching to traverse the crossbar **32**.

When a VC is inserted at the end of a service list, because the VC has sent a cell or because a cell arrived and the VC was not yet in the service list, the timestamp is set to the global time. The global time is provided by a counter that is incremented on each cell time, using a timestamp management method such as those disclosed below. The FARR method preserves the order of cells in a virtual circuit. In an alternative embodiment, each virtual circuit is timestamped with the current time whenever it has a cell sent across the crossbar **32**.

The number of rounds R of matching is flexible, and is to be maximized within the timing and cost constraints of the

input-buffered switch. When no more pairs may be added to the matching, i.e. a maximal matching has been found, no further rounds are necessary. In the exemplary embodiment, R=4. In alternative embodiment, using R=3 rounds is comparable to continuing until a maximal matching is found.

In FIG. 5, the processing of requests in step 44 has the transfer of N extended timestamps for each cell time from each input to the arbiter 30. The timestamp is stored in the arbiter 30, and at most two extended timestamps are sent from each input. The first extended timestamp replaces the extended timestamp of the virtual circuit of a cell that was sent in the last cell time, and the second extended timestamp is that of the virtual circuit of a cell that may have arrived into an empty service list.

In alternative embodiments, first-in-first-out (FIFO) scheduling for voice traffic in the input buffer controllers may be implemented by the FARR method, where the timestamps are associated with entries in a FIFO queue record the time that each cell has arrived. Thus the FARR method performs FIFO queuing across the input buffers at the voice priority level, while providing round-robin service at other priority levels.

In the FARR method disclosed in FIG. 5, the steps 48–50 in the step 46 of arbitration 46 involve selecting a request from a set of contending requests, by picking the request with smallest extended timestamp. Such a selection may be performed using a tree of comparators. Alternatively, each contending request is connected to a shared connection wire for performing an AND function. In step 48, each contending request places the high order bit of their value on the connection wire, so the connection wire has a logic value of 0 exactly when there is a contending request with a high bit of 0. Thus all contending requests that have the high bit set to logic 1 may drop out of the contention.

In step 50 of FIG. 5 the remaining contending requests compete based on their second highest bit, and then on their third highest bit, etc. The number of rounds is the number of bits of resolution in the timestamps, which requires that the extended timestamps being compared be distinct.

Long timestamps require a large amount of storage space in the arbiter 30 and buffer controllers 24–28, and thus slow down the arbitration process in step 46 of FIG. 5. It is preferable to keep the timestamps short, while avoiding wrap-around, since a short timestamp may wrap around frequently in the timestamp counter. In the exemplary embodiment, wrap-around of the timestamp counter is performed as follows: when the global time wraps around, the counter shifts all timestamps right by a single 1 bit, and sets the high-order bit of the global time to 1.

This counter wrap-around method provides that the order of the timestamps is the same as the order of last service of the queues. However, it does require the timestamps to all of the queues be adjusted periodically; for example, once in 512 cell times for 10-bit timestamps. An advantage of this method is that it provides extra bits of timestamp resolution, since only the low order bits are lost when shifting. For example, FARR distinguishes between a VC that has not received service for three global time wraparounds from a VC that has not received service in four global time wraparounds.

An alternative embodiment of the disclosed timestamp management method avoids changing all timestamps simultaneously by keeping longer timestamps in the input buffer controllers 24–28, which are long enough such that wraparound does not occur. The long timestamps may be translated into shorter timestamps before being sent to the arbiter 30, and suitably right-shifted for old timestamps.

A second alternative method that does not require the timestamps to be adjusted after they have been set is the following: let C be a cell with timestamp t, at the head of a service list, and let t_G be the global time. Before using t in the contention process, if the high bit of t_G is 0, flip the high bit of t.

As long as no cell waits in a service list for more than 2^k, where $k=|t_G|-1$ cell times this method provides that the order of the timestamps of contending virtual circuits is exactly the order in which the inputs last sent cells. Therefore the number of bits in the global counter and the timestamps are to be large enough so that the timestamps cycle through each service list in at most 2^k cell times.

A third alternative method is to use an "age" value rather than an actual timestamp. The age of a VC is set to 0 when the VC is inserted at the end of the service list, and is incremented once per cell time. This third alternative method does not employ a global time and hence avoids global time wraparound. Age values are to be kept in the active memory, in which all ages are incremented in parallel.

Alternative exemplary embodiments provide advantages by varying the crossbar speed, such as by running the crossbar switch 32 in FIG. 1 faster than the link speeds. A crossbar switch that runs somewhat faster than the link speed is generally less expensive to implement than an output queued switch. Significant performance improvement is achieved by running the crossbar 10% to 20% faster than the link speed. If the crossbar speed is increased, buffering at the crossbar outputs 34–38 is to be included. In an exemplary embodiment, about 5 cells per link suffices. A 10% crossbar speedup is sufficient for the performance of the disclosed FARR method to be very close to that of PRR.

If the crossbar 32 runs faster than link speed, then the input buffers 18–22 are to be drained faster than link speed. However, this buffer speedup may be avoided by the following exemplary method: for a crossbar 32 running 10% faster than the input buffers 18–22, step 48 in FIG. 5 may be modified to set an input to be ineligible if the input has sent a cell for each of the last 9 time steps. If a 2-cell buffer is then maintained at each input to the crossbar 32, and have cells delayed between 0 and 1 cell times in this 2-cell buffer, then each cell is guaranteed to have arrived from the input buffer by the time the crossbar 32 is to send it.

The disclosed switch 10 implementing FARR may perform in a pipelined manner since the following three steps of the implementation are essentially independent for adjacent cell times and similar in time complexity:

1. contention resolution is performed to determine the set of cells to be sent;
2. the crossbar is set up to send the cells; and
3. the crossbar transmits the cells.

Step (2) above requires most of a cell time, since commercial crossbar chips generally allow only one path to be determined per clock cycle, so the N paths of an N×N crossbar is to be set up in sequence. The crossbar 32 may be programmed in step (2) above while the previous set of cells are being transmitted in step (3) above, since crossbar chips typically have double-buffered control.

SCALING OF THE INPUT-BUFFERED SWITCH

For the input-buffered switch 10 using FARR, scaling may be implemented using the following exemplary specifications:

1. BUFFERS

An ATM cell buffer implementing the input buffers 18–22 is organized in 424 bit words. The throughput of the buffer is limited by the cycle time of the memory array. For DRAM technology with a cycle time of 100 ns, the throughput of

each buffer is about 2.1 Gbits/second, while for SRAM technology with a cycle time of 30 ns the maximum buffer throughput increases to 7.1 Gbits/second.

2. BUFFER CONTROLLERS

The buffer controllers **24–28** keep service lists, perform header translation and address lookup, and manage storage in the buffer. The DATAPATH controller available from AT&T Corp. keeps the service lists in 25 ns SRAM, and performs the pointer manipulation in about 200 ns, well within the 350 ns available at 1.2 Gbits/second. Currently available faster SRAM would easily permit the buffer controllers to manage a 2.4 Gbits/second buffer.

3. CROSSBAR

The size of the crossbar **32** is limited by pin count and wiring constraints. Currently available fast crossbars are available for 32×32 and 64×64 configurations.

4. ARBITRATION

The size of the arbiter **30** is determined by the pins necessary to receive extended timestamps from the buffer controllers. If each extended timestamp is 10 bits long and is passed in a two cycles (or three cycles when the output port number is sent for a newly arrived cell), then 160 pins are required for a 32×32 switch. The same number of pins may be required to return the results of arbitration to the buffer controllers, so a 400 pin package generally suffices. For timing, R=4 rounds of FARR requires 8 steps in sequence, where each step includes finding the maximum of a subset of a row (or column) in the timestamp matrix. Thus at 2.4 Gbits/second there is at most 20 ns available per step. Calculations based on gate delays may be implemented in 0.5 micron CMOS technology. In addition, the arbiter **30** needs to store n^2 timestamps. For a 32×32 switch with 10 bit timestamps, this amounts to 10 Kbits of storage, which is attainable.

Thus with currently available CMOS technology, the total throughput of the disclosed input-buffered switch running FARR is around 80 Gbits/second.

EMPIRICAL EVALUATION

To determine the performance of the disclosed switch 10 implementing FARR, evaluations were performed to compare FARR with the PIM, SLIP, and PRR methods. The SLIP and PIM methods behaved very similarly under evaluation, so the comparisons below refer primarily to PIM, with the understanding that such comparisons apply to SLIP. PRR is evaluated as well since the performance of PRR is to be achieved by FARR to attain output-queued performance in an input-buffered switch.

The following sections present the results of simulating FARR, PIM, and PRR under a variety of work loads, traffic types and switch sizes. In the evaluations a 16×16 switch is used with link speeds of 155 megabits per second.

WORKLOAD MODELING

The performance of the disclosed input-buffered switch **10** implementing FARR is evaluated for different traffic types, voice, video and data. In such evaluations, a voice source generates constant bit rate traffic at a rate of 64 Kbits per second, while a video source generates Poisson traffic at a rate of 1 Mbit per second. As a data source, a cell train model is used in which bursts of cells are interspersed with large gaps. The length of a burst is a geometric random variable with a mean of 12, while the inter-burst gap is chosen uniformly up to some limit, where the limit is such that the average bandwidth is 1 Mbit per second.

The evaluations are performed on a single ATM switch **10**, with the input for this switch **10** generated by setting up a large number of virtual channels. In the evaluations, the input link and the output link for each VC is chosen at random among all the links, subject to the condition that the VC does not enter and leave the same link, and does not increase the bandwidth through any incoming or outgoing link above 95%. Virtual channels are added until the average link load reaches a threshold percentage P. The parameter P, a measure of the switch utilization, is varied in most of the evaluations to determine the behavior of the algorithms under varying switch loads.

Because different virtual channels feeding into the same input may produce cells at the same time, these virtual channels are multiplexed. In a real network this multiplexing is done by earlier switches in the network. In the evaluations PRR schedulers **58–64** are positioned between each input and the VC's feeding into a switch **10** such as VC **66**, as shown in FIG. **6**. The PRR schedulers **58–64** simulate the interleaving of channels by earlier switches in a real network.

In measuring cell delays in the switch **10**, delays inside the PRR schedulers **58–64** at the inputs are not included, since the delay induced by the switch scheduling method inside a switch **10** is to be evaluated.

In addition to the three basic traffic types, one evaluation involves adding a file transfer after the other virtual channels have been set up. A file transfer source uses the same cell train distribution as a data source, but absorbs all the available bandwidth. Its bandwidth is thus the minimum of the remaining bandwidth on the input and output links it uses. The four traffic types are summarized in TABLE 1.

TABLE 1

	Priority	Bandwidth	Distribution
Voice	0	64,000	Constant Bit Rate
Video	1	1,000,000	Poisson
Data	2	1,000,000	Cell Train
File Transfer	2	All Available	Poisson

The workload models are derived by selecting a set of virtual channels each with one of the above traffic source types, using the following models:

- vvd The basic workload model employed, with 10% voice, 40% video and 50% data traffic.
- server A model with non-uniform work load, which may arise in a client-server environment. The first **4** input and output links are connected to servers, while the remaining links are connected to clients. The load on client-client connections is only 10% of the load on server-server or server-client connections. This is implemented by first constructing an instance of the vvd model, then randomly dropping 90% of the client-client virtual channels.
- video A model with 100% Video traffic. This is close to a standard uniform link load model.
- ft After constructing an instance of the vvd model, an additional VC of type File Transfer is added with enough bandwidth to saturate some switch link.

EVALUATIONS

In the evaluations for the above models, the workload model, the switch scheduling method, the switch size, and the utilization parameter P are varied. In each evaluation, the average and maximum queue sizes are measured as well as the average and maximum delay of each traffic type. Each evaluation is run for a 20,000 cell times in order to overcome any initial transient effects. To obtain as much statistical significance as possible, each evaluation is run 50 times. Whenever the average of a parameter is measured; for example, for the delay on voice cells, the average over all

cells in all runs is indicated. Similarly, the maximum value of a parameter indicated is the maximum over all cells in all runs.

By running the evaluations many times over, values are obtained that are representative of all configurations of virtual channels, rather than just a single configuration. In the evaluations presented below, whenever the model is not mentioned it should be understood to be vvd model. Similarly the default switch size is 16×16, and the switch size is indicated when deviating from the default. A logarithmic scale is used when presenting average or maximum queuing delays or queue sizes.

PER-VC FAIRNESS

As shown in FIG. 7, the evaluation shown studies the effect that high-bandwidth virtual channels have on lower bandwidth channels. In FIG. 7 the FARR and PIM methods and the vvd and ft workload models are shown. A difference between the workload models is that ft includes a single file transfer channel that is added after forming a vvd configuration of virtual channels. FIG. 7 shows the average delay on data cells from channels other than the file transfer for PIM and FARR.

When average delays are compared in the vvd model, the file transfer has almost no effect on the maximum queuing delay of other data cells if FARR is used as the switch scheduling method, which demonstrates that FARR insulates virtual channels from the behavior of other virtual channels, as is the case for PRR.

However, the file transfer causes PIM to greatly increase the queuing delay of some data cells in previously existing virtual channels. This is true also of other scheduling algorithms for input-buffered switches, such as SLIP.

PRIORITIES FOR DIFFERENT TRAFFIC TYPES

As shown in FIGS. 8–10, a evaluation is shown illustrating the importance of using multiple priority levels in an ATM switch. Both FARR and PRR respect cell priorities, and therefore have much smaller average queuing delay for voice and video cells than for data cells. In contrast, PIM does not use multiple priority levels, so voice and video cells are subject to large delays under high traffic loads. In addition, PIM has consistently smaller queuing delays than FARR on data traffic. However, this is unavoidable, since FARR is delaying data cells to make way for cells of higher priorities.

UNEVEN TRAFFIC LOADS

It is possible for a switch scheduling method to perform well on uniform traffic loads, but not on the imbalanced traffic loads that frequently arise in real networks. To evaluate FARR on an unevenly distributed traffic load, the server workload model is used including a switch with connections to 4 servers and 12 clients. Traffic from clients to clients is reduced to only 10% of the other traffic types. In this model, FARR exhibits very similar performance to PRR. In FIGS. 11–12 the average and maximum delay on data cells is presented, and other cell types yield similar results.

INPUT BUFFERING MAY YIELD SMALLER MAXIMUM QUEUE SIZE

As shown in FIGS. 13–14, the maximum queue size is an important measure of a switch scheduling method because a scheduling method that has a larger maximum queue size requires larger buffers to avoid dropping cells. In the case that cell loss is tolerable at low levels, the maximum queue size is an indication of the buffer size needed to achieve a low enough level of cell loss probability.

In many of the evaluations, the maximum queue size for input-buffered switches is smaller than that of an output-buffered switch; i.e. a switch with a separate buffer at each output link, since there is "hot spot" contention at an output link in an output-buffered switch. The cells building up in that buffer are being received from a number inputs. In an input-buffered switch, those cells remain at the inputs, and each input buffer holds a smaller number of cells than the output buffer at the overload link.

FIGS. 13–14 show that the maximum queue for FARR is significantly less for FARR than for PRR under two workload models with the vvd model shown in FIG. 13 and the server model shown in FIG. 14.

For a particular workload, if the bottleneck of an input-buffered switch is the switch fabric, rather than an output link, then output-buffering may achieve smaller maximum queue size for that workload.

SCALING WITH SWITCH SIZE

In an N×N switch, output-buffering requires the switch fabric to run at N times the link speed, while input-buffering avoids this speed-up. The advantage of input-buffering increases with larger N, so it is important to determine how well an input-buffered switch schedule scales with increasing N.

FIGS. 15–16 indicate how FARR scales to larger switch sizes, where the average delay of data cells remains constant across different switch sizes, while the maximum delay increases slightly for larger switches.

In a 64×64 switch the maximum delay of a data cell is roughly twice the maximum delay in an 8×8 switch. This correlates to the fact that the evaluation of the 64×64 switch involved 8 times as many cells. When the 8×8 switch is simulated for 8 times as long, the maximum delay observed rose by a factor of 2, so FARR scales well to large switches.

CROSSBAR SPEEDUP

Depending on the relative price of the components of an input-buffered ATM switch, it may be cost-effective to run the crossbar faster than the incoming and outgoing links, giving the switch the advantage of being able to transfer data from inputs to outputs faster than link speed without resorting to a fabric running at N times link speed as in an output-queued switch.

The evaluations show that a crossbar speedup of 10% provides a distinct improvement in the performance of FARR, while a speedup of 20% gives performance which is extremely close to that of PRR. For example, FIG. 17 shows that a crossbar speedup of 10% decreases the average cell delay for data channels by almost a factor of 2 under the vvd workload model, while a 20% speedup gives average delay equal to that of PRR.

NUMBER OF ROUNDS NEEDED BY FARR

The evaluation shown in FIGS. 18–19 investigates the effect of varying R, the number of rounds used by FARR. To maximize switch throughput, it may be required that R be large enough that FARR always finds a maximal matching; i.e. a matching that cannot be extended by the addition of any more pairs. On the other hand, if the link speed is 1 Gigabit per second, there are only 424 nanoseconds available for FARR to choose a matching. For fewer rounds R that FARR uses, FARR is easier to implement.

FIGS. 18–19 show the average delay on data cells and the maximum queue size, respectively, for varying numbers of rounds. The line FARR.mm corresponds to continuing FARR for as many rounds as are needed to produce a maximal matching. As shown in FIGS. 18–19, one round is inadequate, and three rounds provides a substantial improvement for over two rounds. With three rounds the performance is very close to FARR.mm. The evaluations show four rounds to be indistinguishable from FARR.mm. FIGS. 18–19 illustrate that these observations are valid across a

range of switch sizes, so 3 rounds of contention suffice. In a preferred embodiment, 4 rounds provides the best mode for operation of FARR.

PERFORMANCE ON POISSON TRAFFIC

In the evaluations shown in FIGS. **20–21**, the performance of FARR compared to PIM, SLIP and PRR on the video workload model are illustrated. This model consists of setting up video conversations until the average load on the input (or output) links reaches a threshold of P. This is based on presenting data derived from this source since this source is quite close to the uniform Poisson workload model, which has often been used to analyze the performance of switch methods.

FIG. **20** shows that FARR simulates PRR very closely but PIM and SLIP both have maximum cell delays of up to a factor of 8 larger, since different inputs have different numbers of video channels. Methods such as SLIP and PIM that do not exhibit per-channel fairness create large cell delays on the more loaded inputs.

FIG. **21** shows that FARR, PIM and SLIP achieve the same average cell delay up to a threshold of about 80%.

INFLUENCE OF HIGH PRIORITY TRAFFIC ON LOW PRIORITY TRAFFIC

FIGS. **22–24** show worst case behavior of FARR involving low priority traffic being slowed down by large bandwidth higher priority traffic, using an N×N switch shown in FIG. **22** in which each input i is sending a fraction P of the link bandwidth straight across to output i. In addition, INPUT1 has low priority traffic to send to outputs **2**, **3**, . . . , K+1. This traffic pattern is depicted in FIG. **13**.

For the amount of bandwidth given to the low priority traffic, FIG. **23** shows that when K=1, the low priority bandwidth is substantially lower than that achieved by PRR. However, this effect decreases fast as K increases, since PRR has the low priority bandwidth of (1-P) times the link bandwidth. On the other hand, for FARR, the fraction of time that INPUT1 has no high priority cells waiting to send is (1-P), and the fraction of time that one of outputs **2**, **3**, . . . , K+1 has no high priority cell waiting to be sent to it is $(1-P^K)$. Thus the probability that INPUT1 is able to send a low priority cell is $(1-P)(1-P^K)$

FIG. **24** illustrated the effect of crossbar speedup for the worst case of this phenomenon, when K=1. A speedup of 10% returns half of the lost low priority bandwidth, while a speedup of 20% is indistinguishable from PRR.

While the disclosed input-buffered switch and FARR scheduling method has been particularly shown and described with reference to the preferred embodiments, it will be understood by those skilled in the art that various modifications in form and detail may be made therein without departing from the scope and spirit of the invention. Accordingly, modifications such as those suggested above, but not limited thereto, are to be considered within the scope of the invention.

What is claimed is:

1. An apparatus for scheduling input buffers of fixed size packet switches, including asynchronous transfer mode (ATM) switches, having input and output ports, the apparatus comprising:

a plurality of input buffers having associate virtual circuits for receiving the arriving cells, each input buffer associated with a respective input port;

a cell switching fabric for processing the received cells from the input buffers to the output ports; and

a scheduling control circuit which uses timestamps associated with the virtual circuits for controlling the processing of the received cells through the cell switching fabric using fair arbitration round robin (FARR) to allow each virtual circuit to receive a fair share of bandwidth.

2. An apparatus for scheduling input buffers of fixed size packet switches, including asynchronous transfer mode (ATM) switches, having input ad output ports, the apparatus comprising:

a plurality of input buffers for receiving arriving cells, each input buffer is associated with a respective input port and includes a service list associated with each priority level for each output port;

a cell switching fabric for processing the received cells from the input buffers to the output ports;

a plurality of queues, each queue associated with a virtual circuit connecting the input buffer with at least one output port and having an associated timestamp; and

a scheduling control circuit which uses the timestamps to generate a matching of input buffers to output ports to control the processing of the received cells through the cell switching fabric using a fair arbitration round robin (FARR) program.

3. A method for scheduling input buffers of fixed size packet switches, including asynchronous transfer mode (ATM) switches, having input and output ports, the method comprising the steps of:

receiving cells in the input buffers;

pre-selecting at each input buffer a virtual circuit for each output port;

performing request processing to send a timestamp associated with each pre-selected virtual circuit to a scheduling control circuit;

generating a matching of the input buffers to the output ports from the scheduling control circuit using the timestamps; and

processing the received cells through a cell switching fabric to the output ports using the matching.

4. The method of claim **3** wherein the step of matching includes the step of using an arbiter unit operating a fair arbitration round robin (FARR) program.

5. The method of claim **3** wherein the step of generating a matching includes the steps of:

granting, for each unmatched output port, the request having a minimum timestamp;

accepting, for each unmatched input buffer, the granted request with a minimum timestamp; and

incorporating the accepted granted request in the matching.

6. The method of claim **5** wherein the steps of granting requests, accepting granted requests, and incorporating the accepted granted requests are performed for a predetermined number of rounds.

7. The method of claim **3** including the step of using timestamps and a central arbiter to fairly arbitrate between input buffers in an input-buffered switch.

8. The method of claim **3** including the step of using timestamps and a central arbiter to fairly arbitrate between ports in an input-buffered switch.

9. The method of claim **3** including the step of using timestamps and a central arbiter to fairly arbitrate between virtual circuits in an input-buffered switch.

10. The apparatus of claim **1** wherein the scheduling control circuit pre-selects at each input buffer a respective virtual circuit for each output port.

11. The apparatus of claim 1 further comprising:

means for pre-selecting at each input buffer a respective virtual circuit for each output port; and

means for performing request processing to send a respective timestamp associated with each pre-selected virtual circuit to the scheduling control circuit.

12. The apparatus of claim 1 further comprising:

an arbiter unit operating a FARR program to process the received cells.

13. The apparatus of claim 1 further comprising:

the scheduling control circuit which uses the timestamps to generate a matching of input buffers to output ports, wherein each unmatched output buffer is associated with a respective input buffer as an unmatched input buffer;

means for pre-selecting at each input buffer a virtual circuit for each output port;

means for performing request processing on at least one request to send a timestamp associated with each pre-selected virtual circuit to the scheduling control circuit;

means for granting, for each unmatched output port, the request having a minimum timestamp;

means for accepting, for each unmatched input buffer, the granted request with a minimum timestamp; and

means for incorporating the accepted granted request in the matching.

14. The apparatus of claim 1 further comprising:

a central arbiter using the timestamps to fairly arbitrate between input buffers in an input-buffered switch.

15. The apparatus of claim 1 further comprising:

a central arbiter using the timestamps to fairly arbitrate between ports in an input-buffered switch.

16. The apparatus of claim 1 further comprising:

a central arbiter using the timestamps to fairly arbitrate between the virtual circuits in an input-buffered switch.

17. The apparatus of claim 2 wherein the scheduling control circuit pre-selects at each input buffer the virtual circuit for each output port.

18. The apparatus of claim 2 further comprising:

means for pre-selecting at each input buffer the virtual circuit for each output port; and

means for performing request processing to send a respective timestamp associated with each pre-selected virtual circuit to the scheduling control circuit.

19. The apparatus of claim 2 further comprising:

an arbiter unit operating the FARR program to process the received cells.

20. The apparatus of claim 2 further comprising:

means for performing request processing on at least one request to send a timestamp associated with each circuit to the scheduling control circuit;

means for granting, for each unmatched output port, the request having a minimum timestamp;

means for accepting, for each unmatched input buffer, the granted request with a minimum timestamp; and

means for incorporating the accepted granted request in the matching.

21. The apparatus of claim 2 further comprising:

a central arbiter using the timestamps to fairly arbitrate between input buffers in an input-buffered switch.

22. The apparatus of claim 2 further comprising:

a central arbiter using the timestamps to fairly arbitrate between ports in an input-buffered switch.

23. The apparatus of claim 2 further comprising:

a central arbiter using the timestamps to fairly arbitrate between the virtual circuits in an input-buffered switch.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 5,517,495

DATED : May 14, 1996

INVENTOR(S) : Carsten Lund, Steven Phillips, Nicholas F. Reingold

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 11, claim 1, line 62, delete "associate" and insert --associated--.

Column 12, claim 2, line 9, delete "ad" and insert --and--.

Signed and Sealed this

Twenty-seventh Day of March, 2001

Attest:

Nicholas P. Godici

NICHOLAS P. GODICI

Attesting Officer

Acting Director of the United States Patent and Trademark Office

US006545979B1

(12) **United States Patent**
Poulin

(10) **Patent No.:** **US 6,545,979 B1**
(45) **Date of Patent:** **Apr. 8, 2003**

(54) **ROUND TRIP DELAY MEASUREMENT**

(75) Inventor: **André Poulin**, Hull (CA)

(73) Assignee: **Alcatel Canada Inc.**, Kanata (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/200,442**

(22) Filed: **Nov. 27, 1998**

(51) **Int. Cl.**[7] **H04L 12/26**; H04J 3/14

(52) **U.S. Cl.** **370/241.1**; 370/241; 370/248; 370/249; 370/250; 370/389; 370/395.1; 370/905; 709/224; 714/47

(58) **Field of Search** 370/241, 241.1, 370/248, 249, 250, 389, 395.1, 905; 709/223, 224; 714/1, 47

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,260,935 A	*	11/1993	Turner	370/60
5,339,311 A	*	8/1994	Turner	370/60
5,471,631 A	*	11/1995	Beardsley et al.	395/650
5,483,523 A	*	1/1996	Nederlof	370/58.3
5,521,907 A	*	5/1996	Ennis, Jr. et al.	370/17
5,793,976 A	*	8/1998	Chen et al.	395/200.54
5,812,528 A	*	9/1998	VanDervort	370/235
6,023,455 A	*	2/2000	Takahashi	370/249
6,097,699 A	*	8/2000	Chen et al.	370/231

OTHER PUBLICATIONS

ITU–T I.610, Integrated Services Digital Network (ISDN), Maintenance Principles, "B–ISDN Operation and Maintenance Principles and Functions", 11/95.

The ATM Forum Technical Committee, Traffic Management Specification, Version 4.0, Apr. 1996.

ITU–T I.356, Series I: Integrated Services Digital Network, Overall network aspects and funtions—Performance objectives, "B–ISDN ATM layer cell transfer performance", 10/96.

ITU–T, O.191, Series O: Specifications of Measuring Equipment, Equipment for the measurement of digital and analogue/digital parameters, "Equipment to access ATM layer cell transfer performance", 04/97.

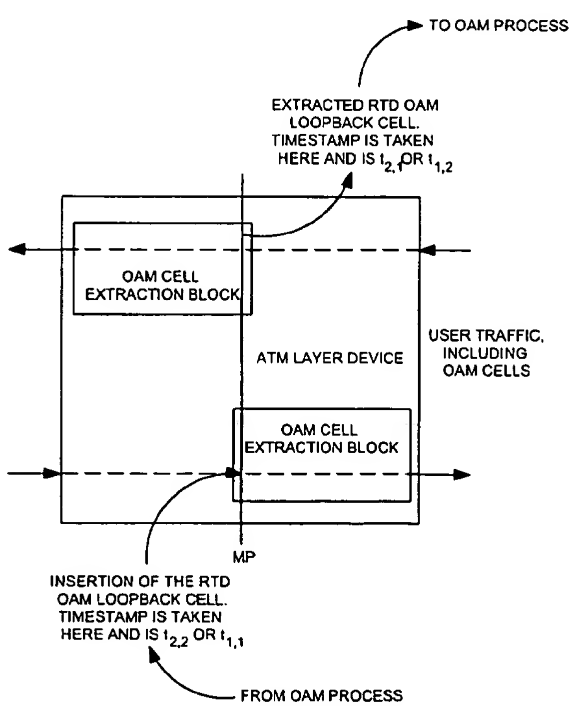
* cited by examiner

Primary Examiner—Hassan Kizou
Assistant Examiner—Joe Logsdon
(74) *Attorney, Agent, or Firm*—Marks & Clerk

(57) **ABSTRACT**

A system and method for calculating round trip delay (RTD) values in a switched digital network such as an asynchronous transfer mode (ATM) network. A loopback cell such as an ATM operation and maintenance (OAM) cell is used to carry a delta value through the network. The delta value, which represents a processing interval at a loopback node or an intermediate node, is calculated utilizing timestamps generated at specific ingress and egress ports of network nodes. Cell Transfer Delay (CTD) and Cell Delay Variations (CDV) values are calculated based on the measured RTD.

10 Claims, 4 Drawing Sheets



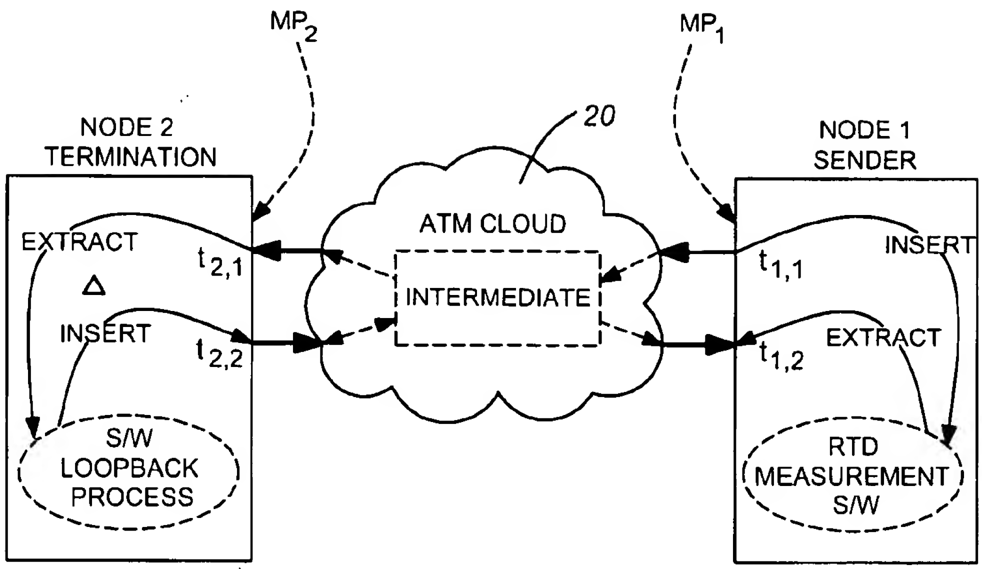

MP2
MP1
20
NODE 2
TERMINATION
NODE 1
SENDER
ATM CLOUD
EXTRACT
t2,1
Δ
INSERT
t2,2
INTERMEDIATE
t1,1
INSERT
EXTRACT
t1,2
S/W
LOOPBACK
PROCESS
RTD
MEASUREMENT
S/W

FIG. 1

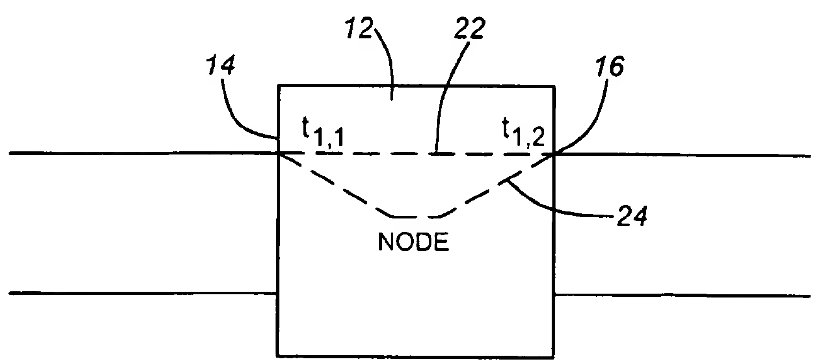

12
22
14
16
t1,1
t1,2
24
NODE

FIG. 5

Byte	CONTENT	
1	ATM HEADER Byte 1	
2	ATM HEADER Byte 2	
3	ATM HEADER Byte 3	
4	ATM HEADER Byte 4	
5	ATM HEADER Byte 5	
6	OAM Cell Type	OAM Function Type
7	Loopback indication bit	
8	Correlation Tag Byte 1	
9	Correlation Tag Byte 2	
10	Correlation Tag Byte 3	
11	Correlation Tag Byte 4	
11	Loopback Location ID (optional) Byte 1	
12	Loopback Location ID (optional) Byte 2	
13	Loopback Location ID (optional) Byte 3	
14	Loopback Location ID (optional) Byte 4	
15	Loopback Location ID (optional) Byte 5	
16	Loopback Location ID (optional) Byte 6	
17	Loopback Location ID (optional) Byte 7	
18	Loopback Location ID (optional) Byte 8	
19	Loopback Location ID (optional) Byte 9	
20	Loopback Location ID (optional) Byte 10	
212	Loopback Location ID (optional) Byte 11	
23	Loopback Location ID (optional) Byte 12	
24	Loopback Location ID (optional) Byte 13	
25	Loopback Location ID (optional) Byte 14	
26	Loopback Location ID (optional) Byte 15	
27	Loopback Location ID (optional) Byte 16	
28	Source ID (optional) Byte 1	
29	Source ID (optional) Byte 2	
30	Source ID (optional) Byte 3	
31	Source ID (optional) Byte 4	
32	Source ID (optional) Byte 5	
33	Source ID (optional) Byte 6	
34	Source ID (optional) Byte 7	
35	Source ID (optional) Byte 8	
36	Source ID (optional) Byte 9	
37	Source ID (optional) Byte 10	
38	Source ID (optional) Byte 11	
39	Source ID (optional) Byte 12	
40	Source ID (optional) Byte 13	
41	Source ID (optional) Byte 14	
42	Source ID (optional) Byte 15	
43	Source ID (optional) Byte 16	
44	Unused (6AH) byte 1 for conventional OAM loopback cells---t_3 byte 1 for RTD loopback cell	
45	Unused (6AH) Byte 2 for conventional OAM loopback cells---t_3 byte 2 for RTD loopback cell	
46	Unused (6AH) Byte 3 for conventional OAM loopback cells---t_3 byte 3 for RTD loopback cell	
47	Unused (6AH) Byte 4 for conventional OAM loopback cells---t_3 byte 4 for RTD loopback cell	
48	Unused (6AH) Byte 5 for conventional OAM loopback cells----- Unused (6AH) Byte 1 for RTD loopback cell	
49	Unused (6AH) Byte 6 for conventional OAM loopback cells----- Unused (6AH) Byte 2 for RTD loopback cell	
50	Unused (6AH) Byte 7 for conventional OAM loopback cells----- Unused (6AH) Byte 3 for RTD loopback cell	
51	Unused (6AH) Byte 8 for conventional OAM loopback cells----- Unused (6AH) Byte 4 for RTD loopback cell	
52	CRC-10 Byte 1	
53	CRC-10 Byte 2	

FIG. 2

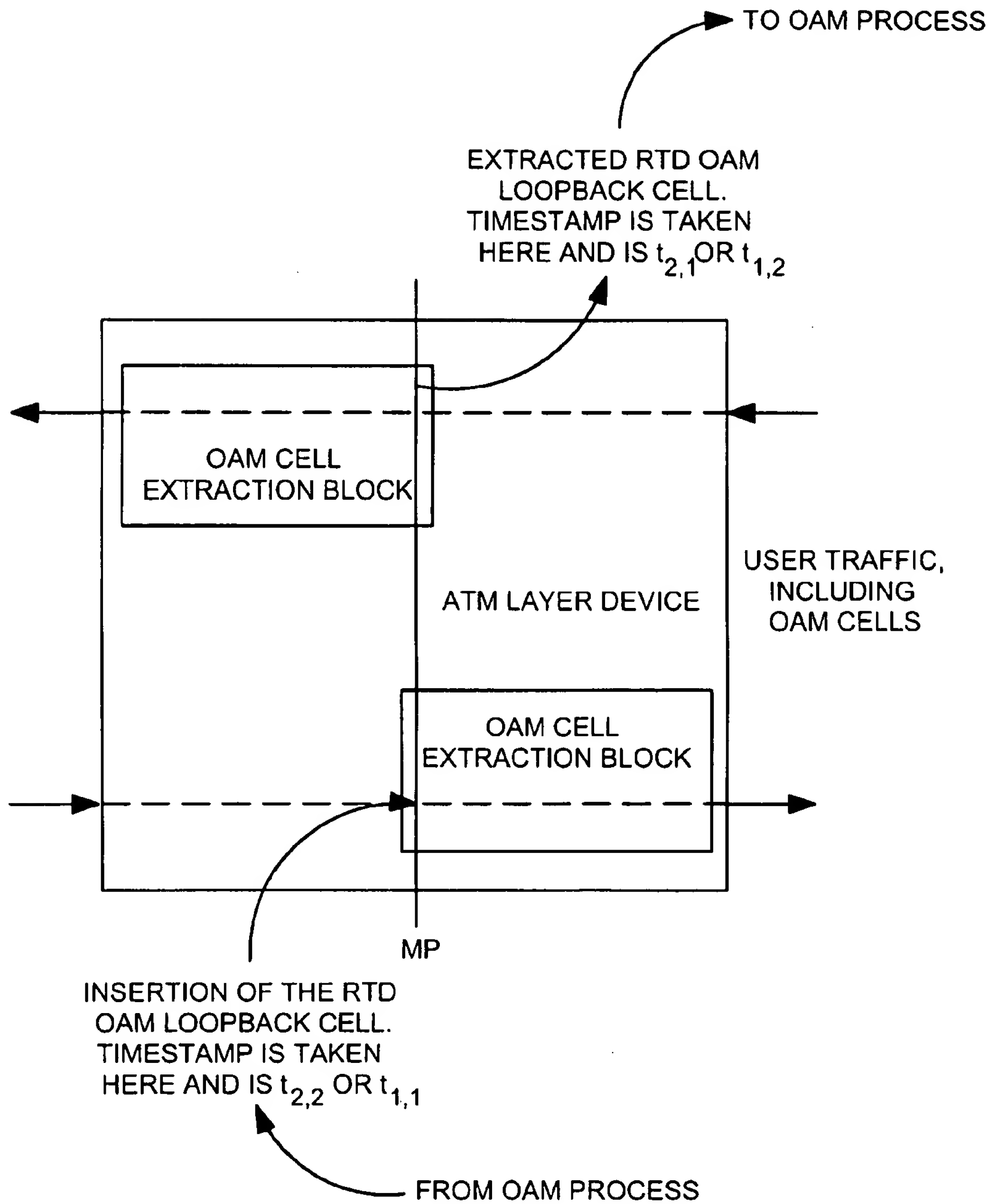

TO OAM PROCESS
EXTRACTED RTD OAM LOOPBACK CELL. TIMESTAMP IS TAKEN HERE AND IS $t_{2,1}$ OR $t_{1,2}$
OAM CELL EXTRACTION BLOCK
ATM LAYER DEVICE
USER TRAFFIC, INCLUDING OAM CELLS
OAM CELL EXTRACTION BLOCK
MP
INSERTION OF THE RTD OAM LOOPBACK CELL. TIMESTAMP IS TAKEN HERE AND IS $t_{2,2}$ OR $t_{1,1}$
FROM OAM PROCESS

FIG. 3

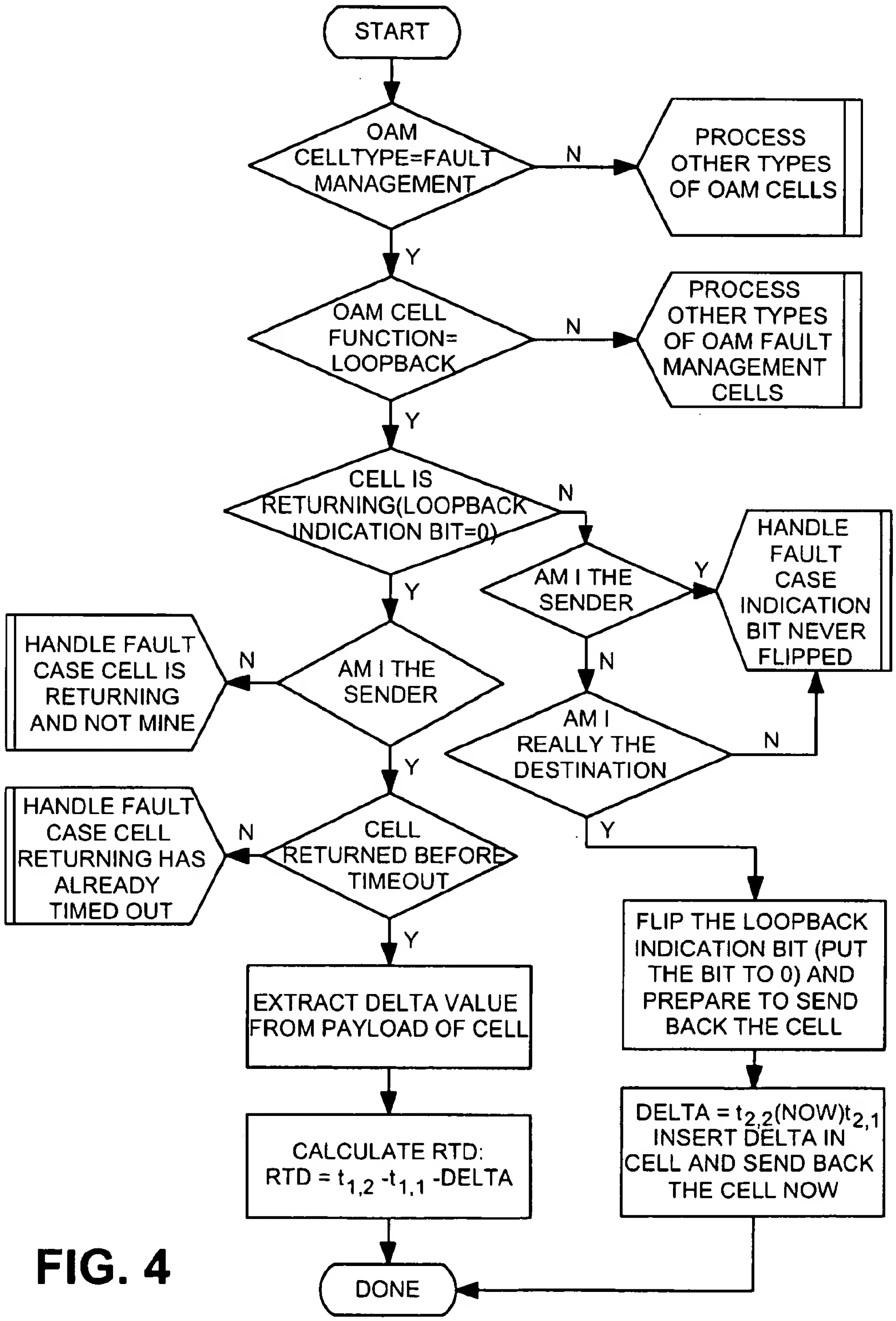

START
OAM CELLTYPE=FAULT MANAGEMENT
N
PROCESS OTHER TYPES OF OAM CELLS
Y
OAM CELL FUNCTION= LOOPBACK
N
PROCESS OTHER TYPES OF OAM FAULT MANAGEMENT CELLS
Y
CELL IS RETURNING(LOOPBACK INDICATION BIT=0)
N
AM I THE SENDER
Y
HANDLE FAULT CASE INDICATION BIT NEVER FLIPPED
N
AM I REALLY THE DESTINATION
N
Y
FLIP THE LOOPBACK INDICATION BIT (PUT THE BIT TO 0) AND PREPARE TO SEND BACK THE CELL
DELTA = $t_{2,2}$(NOW)$t_{2,1}$ INSERT DELTA IN CELL AND SEND BACK THE CELL NOW
Y
AM I THE SENDER
N
HANDLE FAULT CASE CELL IS RETURNING AND NOT MINE
Y
CELL RETURNED BEFORE TIMEOUT
N
HANDLE FAULT CASE CELL RETURNING HAS ALREADY TIMED OUT
Y
EXTRACT DELTA VALUE FROM PAYLOAD OF CELL
CALCULATE RTD: RTD = $t_{1,2}$ -$t_{1,1}$ -DELTA
DONE

FIG. 4

ROUND TRIP DELAY MEASUREMENT

FIELD OF THE INVENTION

This invention relates to packet and cell switched digital networks and more particularly to a system and method for determining the time interval associated with processing a cell at a network node and for calculating the round trip delay characteristics of traffic between end systems employing a time stamp insertion technique.

BACKGROUND

Packet switching systems, such as asynchronous transfer mode (ATM), typically rely on switching nodes within a network to transport digital traffic from a source to a destination or from end system to end system. Such systems are frequently operated by service providers who undertake to deliver digital traffic in accordance with a negotiated service agreement. In order to evaluate quality of service (QoS) performance it is frequently desirable to monitor certain service related criteria such as cell transfer delay (CTD) and cell delay variations (CDV). This requires that there be an accurate evaluation of the times involved in transporting data traffic through the network. Typically, delays can occur in the network at the physical layer, at the switching layer or through queuing at switching nodes.

One know solution to measuring delays in digital traffic relates to ATM systems in which Performance Monitoring (PM) cells are used to calculate Cell Delay Variations (CDVs). The known techniques for CDV calculations based on PM cells, however, require that the measuring points (MPs) have a synchronized clock. In any event, the reliability of PM cells to measure delay is still under study.

The ATM Forum Traffic Management Specification version 4.0 includes reference to a loopback cell for use in monitoring specific system parameters. This loopback cell, known herein as an Operations and Maintenance (OAM) cell, is transmitted, on demand, by one of the end systems, e.g. the source, through the various intermediate switching elements in the network to the opposite end system, e.g. the destination, and reports back to the source with relevant information. This information includes on-demand connectivity status, instances of fault detection and in the event of a fault detection, information on the fault location, and pre-service connectivity verification. An OAM cell, in an ATM environment has a fixed length of 53 bytes including a 5 byte header as will be discussed in greater detail later. Advantageously, there are byte location assignments in an OAM cell that have not heretofore been used. In accordance with the present invention these previously unassigned byte locations are used for on-demand round trip delay measurement information.

SUMMARY OF THE INVENTION

Accordingly, the present invention provides a method and system for determining the Round Trip Delay (RTD) value which can be used to calculate CTD and CDV. The method and system makes use of the aforementioned loopback OAM cell in the measuring of cell transfer delay in a round trip fashion and to record times required to process cells at network elements. The system and method employ the concept of time stamps. Advantageously, implementation of the invention does not require a synchronization clock.

Therefore in accordance with a first aspect of the present invention there is provided in a packet switched digital

network having at least one network element a system for calculating processing intervals at the network element comprising: first means to detect an arrival of a selected packet at the element; first timing means to generate a first time stamp upon detecting the arrival of the packet; second means to note the instance of sending the selected packet from the element; second timing means to generate a second time stamp upon detecting the sending; and means to obtain a processing interval by subtracting the first time stamp from the second time stamp.

In a preferred embodiment of this aspect of the invention the selected cell is a loopback OAM cell which is transmitted, on demand, by an end system through the network and back to the end system.

In this preferred embodiment the network element records the processing interval also known as the delta value in the OAM cell. It is within the scope of the present invention for the OAM cell to record accumulated delta values occurring due to the OAM cell being processed by a number of network elements.

In accordance with a second aspect of the invention there is provided a method of determining a processing interval of a cell at a network element in a packet switched digital network. The method comprises: recording a first time stamp upon arrival of a selected cell at the network element; recording a second time stamp upon sending the selected cell from the element; and obtaining a process interval or delta value by subtracting the value of the first time stamp from the value of the second time stamp.

The present invention also contemplates the calculation of a round trip delay wherein a time stamp is attached to a loopback OAM cell upon being sent from a source, a second time stamp is generated upon arrival at a destination, a third time stamp is generated upon the cell being returned by the destination and a fourth time stamp is generated upon receipt at the source. The delta value, which is the difference between the third time stamp and the second time stamp, is calculated at the destination and recorded in the OAM cell. At the source the round trip delay is determined by subtracting the first time stamp from the fourth time stamp and then subtracting therefrom the delta value.

Therefore, in accordance with a third aspect of the present invention there is provided a method of determining the round trip delay of digital traffic from a source node to a termination node and back to the source through a switched digital network, the method comprising: recording a first time stamp when a selected cell is transmitter by the source node; recording a second time stamp upon receipt of the cell by the termination node; recording a third time stamp when the cell is returned by the termination node; calculating delta by subtracting the second time stamp from the third time stamp; marking the selected cell with the delta value; recording a fourth time stamp upon receipt of the cell by the source node; and determining the round trip delay by subtracting the first time stamp from the fourth time stamp and subtracting delta from the result.

In a preferred embodiment of this aspect the selected cell is an OAM cell. Further, interrupt means are preferably used by the termination node and/or the source node to detect the arrival of the OAM cell.

According to a further aspect of the invention there is provided a system for determining the round trip delay of digital traffic between a source node and a termination node in a switched digital network, the system comprising: first means to record a first time stamp upon transmission of a selected cell by the source node; second means to record a

second time stamp upon receipt of the selected cell by the termination node; third means to record a third time stamp and to calculate a delta value by subtracting the second time stamp from the third time stamp; marking means to mark the selected cell with the delta value; fourth means to record a fourth time stamp upon receipt of the selected cell by the source node; and means to calculate the round trip delay by subtracting the first time stamp from the fourth time stamp and subtracting the delta value from the remainder.

In a preferred embodiment of this aspect of the invention the source node and/or termination nodes include interrupt means to ensure prompt detection of the arrival of the selected cell. The selected cell is preferably an OAM cell.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in greater detail with reference to the attached drawings wherein:

FIG. **1** is a high level diagram of a sender node and a termination node showing round trip delay measurement parameters;

FIG. **2** shows a loopback cell structure according to the invention;

FIG. **3** illustrates an OAM cell flow at the ATM device layer;

FIG. **4** is a flow diagram of the processing of an OAM loopback cell with RTD calculations; and

FIG. **5** shows an intermediate switching node illustrating different paths for OAM cells and user data cells with receive and send time stamp indications for OAM cells.

DETAILED DESCRIPTION OF THE INVENTION

In constant bit rate (CBR) ATM traffic used for telephony purposes, the service contract will typically specify that the one way delay must be smaller than 30 msec in order for the system to work without an echo cancellation device. One method of determining the one way delay is to use the round trip delay which in this example involves measuring the total time it takes for cells to go to the far end and come back. Assuming that the delay in a downstream connection is equal to the delay in an upstream connection, then the one way delay can be obtained by dividing the RTD by 2. In the above example, the one way delay would, in most cases, be smaller than 30 msec anyway.

In the present invention the round trip delay is calculated by measuring the time it takes for a designated cell to travel between end systems including time spent at intermediate nodes. The round trip delay value will typically include all queuing, switching and routing delays from the one end system, e.g. the sending node, to the other end system, e.g. termination node, and then from the termination node back to the sending node. In FIG. **1** this includes delays from MP_1 to MP_2 and then from MP_2 to MP_1. In this example any delay before MP_1 and after MP_2 is not included in the Round Trip Delay calculation. Thus at node 2, any software (or hardware) introduced delay is not part of the Round Trip Delay and must be excluded from the equation. As will be discussed later the designated cell, also known as a round trip delay cell, takes a different path, particularly at the termination node, than does a user data cell and therefore will require special consideration. Node 2 will need to calculate how much time was spent by software (and hardware if not negligible) for the processing of the round trip delay cell and must have a way to tell Node 1 about this time.

The solution is to calculate and send back in the designated or special cell the time spent by the software and/or hardware in the termination node to process the cell. For the purpose of this calculation the present invention relies on timestamps which are recorded in the cell at specific points in the system.

The specific or designated cell identified above and used in the present invention is a standard OAM loopback cell. In FIG. **1** an OAM cell is launched from the point where the RTD is calculated (node 1), traverses one or more intermediate ATM switches **20** and is received by the terminating point (node 2).

The terminating point (node 2) then processes that cell and returns it back to the sender (node 1) with the indication bit flipped (telling the sender that this cell was well received and sent back). Since the terminating point took some time to process that cell, that processing interval is inserted in the OAM loopback cell (to be described later with reference to FIG. **2**) before it is sent back.

The original sender of the cell then receives the cell, with the delay introduced by the terminating point, and can then calculate the time it took for the cell to leave and come back, excluding the processing time at the looping end.

The purpose of the RTD is to calculate the time spent by a cell in the ATM network (physical layer, switching layer, queuing, etc..) and not the time for the cell to be looped back at the terminating point. This is the reason for the introduction of processing time or a delta value in a standard OAM loopback cell by the present invention.

In order to be able to measure the RTD, a standard F4 (VP) or F5 (VC) OAM loopback cell is used. This standard OAM loopback cell is currently used in most available ATM switches and is used to verify connectivity at setup time or in service. It is also used for fault localization when a connection is failing. An additional purpose for the OAM cell, as introduce by the present invention, is for the insertion of the delta value for round trip delay calculation purposes. The round trip delay is calculated with the help of time stamping and the aforementioned delta value.

FIG. **2** illustrates the structure of the OAM loopback cell. At present this type of cell has 8 octets that are unused. Unused OAM cell information field octets are coded as 0X6A. The unused octet fields are, typically, not checked by the receiver of an OAM cell, thus providing the ability to enhance those OAM cells. Backward compatibility is thus preserved because node elements which do not support the RTD measurement will process the cell normally i.e. the cell is not discarded simply because the Hex values 0X6A were not found in the unused octets. For example, if Node 2 (FIG. **1**) receives an OAM loopback cell from an originating node (e.g. Node 1) which does not support RTD calculations, Node 2 would still introduce the delta value in the returning cell and Node 1, upon receipt of the returning cell, would still consider the cell to be valid. In other words, even if a delta value is in the cell, Node 1 will process the cell as a standard OAM loopback cell.

The RTD measurement according to the invention uses the first four unused octets (Byte no. **44**,**45**,**46** and **47**) of a standard OAM loopback cell structure as can be seen in FIG. **2**. Those four bytes will be used to store a "delta value" which was inserted by the termination point at MP_2. The "delta value" is referred to as t_3 in this document.

The resolution of all time stamps and for the delta value is in μsec. For the delta value (32 bit), the valid range will be 0 to 0X6A6A6A69 because of the 0X6A6A6A6A value found in a standard OAM loopback cell. This means that if

the value 0X6A6A6A6A is received in the delta value the looping end does not support RTD measurements.

Conventionally, if an OAM loopback cell is not returned to the sender node within five seconds from the send time, it is considered lost. This gives a practical limit to the delta value of five seconds. Thus the valid range for t_3 (delta value) is 0 to 5 seconds. Although five seconds could have fit in a 3 byte value, 4 bytes are used in the preferred embodiment because it is handled more easily by control processors and it is also expandable.

Using a standard OAM F4 and F5 type of cell for calculating the round trip delay value gives a method of performing a totally non-intrusive measurement. OAM RTD cells will not be sent more than one cell per five seconds to follow the OAM loopback standards and thus does not affect the bandwidth needed.

Using FIG. 1 the round trip delay measurement time stamping process will now be described in more detail. FIG. 1 shows the overall setup of a connection. The connection can be a VP (F4) or VC (F5) level connection.

Sequence of events:

1) Node 1 receives a request to measure the round trip delay on a specific connection. The entity at point MP_1 takes a first time stamp ($t_{1,1}$) just before transmitting the OAM loopback cell towards the ATM cloud.
2) The cell travels through the ATM cloud and arrives at point MP_2. Point MP_2 takes a second time stamp ($t_{2,1}$) when the OAM loopback cell arrives (this process may be interrupt driven to improve accuracy in the measurement process).
3) At Node 2, software and/or hardware processes the OAM loopback cell, and takes a third time stamp ($t_{2,2}$) just before sending back the cell. Now the delta value will be calculated in the following fashion.

$$\text{Delta } (t_3)=t_{2,2}-t_{2,1}$$

The delta value is put in the first four unused bytes of the cell shown in FIG. 2 to be returned to node 1. The cell travels back through the ATM cloud towards Node 1 and arrives at point MP_1. Point MP_1 takes a fourth time stamp ($t_{1,2}$) when the OAM loopback cell arrives (this process is again preferably interrupt driven for better accuracy in the measurement process). The total Round Trip Delay can now be calculated at Node 1 as follows:

round trip delay=(Time when the cell was received)−(Time when the cell was sent)−Delta value

i.e., $RTD=t_{1,2}-t_{1,1}-t_3$

ITU-T Recommendation I.356 "B-ISDN ATM Layer Cell Transfer Performance", October 1996 provides details on how both CDV and CTD are calculated utilizing the RTD value.

FIG. 3 illustrates the OAM cell flow at the ATM device layer. This is a general, simplified block diagram of an ATM device layer used to extract and insert OAM cells. All cells going through this device are verified and OAM cells marked to be extracted are put aside and the control processor is notified by an interrupt sequence. The control processor then takes a time stamp and extracts the cell. OAM cells to be inserted in the device are copied to the device and the device is then triggered to send the cell.

Depending on location i.e., Node 1 or Node 2, different processing is performed. If at Node 1, then a time stamp is taken when the cell is sent and when the cell is received. Those time stamps are stored away for final calculation of RTD. If at Node 2, a time stamp is taken when the cell is received. When the cell is ready to be transmitted, a second time stamp is taken and the delta value is calculated and inserted in the cell. The cell is then transmitted.

FIG. 4 shows a flow chart of the processing of an OAM Loopback cell. This flowchart is simplified and only illustrates the case of a loopback cell with an included RTD delta value.

As the OAM Loopback cell is propagated on a specific virtual connection (VC) or virtual path (VP) through the ATM network in the forward and return directions, various switching nodes traversed by the OAM cell may cause it to deviate from the VC for internal processing purposes. In other words, the OAM cell may not follow the same VC or VP route as typical data cells within the ATM switch. This deviation from the internal data path might introduce unwanted delays in the round trip time that is being measured.

Thus, as an alternative, the OAM Loopback cell can carry a cumulative delay delta and any switching node that causes the OAM cell to deviate from the internal path followed by data cells for the particular VC or VP adds its computed delta to the value already in the OAM Loopback cell. The RTD measurement in this embodiment is computed at the source node in the same fashion as described above.

FIG. 5 represents such an intermediate switching element or node 12 in a network environment. Each such element has an ingress port **14** and an egress port **16**. Within the element, user data cells may be routed on a different VC or VP than the OAM loopback cell. This is shown in FIG. **5** by the two paths **22** and **24** respectively. Thus, when an OAM cell takes a different route than user data cells through a switching node the processing interval in that node needs to be considered. In accordance with the present invention a first time stamp $t_{1,1}$ is generated upon an OAM cell arriving at a switching node and a second time stamp $t_{1,2}$ is generated upon sending of the OAM cell from the node. The difference between the first time stamp and the second time stamp is the processing interval and is marked in the OAM cell for use in determining time related information such as round trip delay.

While certain specific embodiments of the invention have been described and illustrated it will be apparent to one skilled in the art, to which the invention pertains, that numerous alternatives and/or variations can be implemented without departing from the basic concept. It is to be understood, however, that such alternatives and/or variations will fall within the full scope of the invention as defined by the appended claims.

What is claimed is:

1. A system for calculating a processing interval at a network element in a packet switched digital network, said system comprising: first detection means at an ingress port of said network element to detect an arrival of a designated loopback cell; first timing means to generate a first time stamp value upon detecting the arrival of the designated loopback cell; second detection means at an egress port of said network element to detect the sending of said designated loopback cell from said network element; second timing means to generate a second time stamp value upon detection of the sending of the designated loopback cell; and processing means to calculate a processing interval by subtracting said first dime stamp value from said second time value, and to store said processing interval in said designated loopback cell for transmission to a downstream network element, the calculation of said processing internal being independent of a network synchronization clock.

2. A system as defined in claim 1 wherein said switched digital network is an asynchronous transfer mode (ATM) network.

3. A system as defined in claim 2 wherein said loopback cell is an ATM operations and maintenance (OAM) cell.

4. A system as defined in claim 3 having interupt means to detect the arrival of said designated packet.

5. A method of determining a processing interval of a designated loopback cell at a network element in a packet switched digital network, the method comprising: recording a first time stamp upon arrival of said designated loopback cell at said network element; recording a second time stamp upon sending said designated loopback cell from said network element; obtaining a process interval by subtracting the value of said first time stamp from the value of said second time stamp; and storing the processing interval in said designated loopback cell for transmission to a downstream network element, the processing interval being obtained independent of a network synchronization clock.

6. A method of calculating a round trip delay between a sending node and a termination node in a communications system employing a loopback OAM cell, said method comprising: recording a first time stamp when said loopback OAM cell is sent from said sending node, recording a second time stamp upon arrival of said loopback OAM cell at said termination node, recording a third time stamp upon said loopback OAM cell being sent back by said termination node and recording a fourth time stamp upon receipt of said loopback OAM cell at said sending node source; calculating a delta value by subtracting said second time stamp from said third time stamp; storing said delta value in said loopback OAM cell for transmission to said termination node and, calculating said round trip delay by subtracting said first time stamp from said fourth time stamp and subtracting said stored delta value from the remainder, said delta value being calculated independent of a network synchronization clock.

7. The method as defined in claim 6 wherein said round trip delay is used to calculate a cell delay variation (CDV) respecting traffic in said digital network.

8. The method as defied in claim 6 wherein said round trip delay value is used to calculate cell transfer delay (CTD) respecting traffic in said digital network.

9. A system for calculating cell transfer delay (CTD) in an asynchronous transfer mode (ATM) communications network utilizing a round trip delay (RTC) value respecting an Operations and Maintenance (OAM) loopback cell sent from a sender node to a termination node and returned to said sender node, said system comprising: a first timing means at said sender node for recording a first time stamp representing transmission time of said OAM cell; a second timing means at said termination node for recording a second time stamp representing receipt of said OAM cell; a third timing means at said termination node for recording a third time stamp representing retransmission of said OAM cell toward said sender node; delta calculating means in said termination node for calculating a processing interval at said termination node by subtracting said second time stamp from said third time stamp; means at said termination node for storing said calculated processing interval in said OAM cell; fourth timing means at said sender node for recording a fourth time stamp representing receipt of said OAM cell by said sender node; calculating means for determining round trip delay value by subtracting said first time stamp from said fourth time stamp and subtracting said stored processing interval from said remainder; and means to calculate CTD utilizing said RTD value, said round trip delay value being calculated independent of a network synchronization clock.

10. A system as defined in claim 9 further including means to calculate cell delay variation (CDV) utilizing said RTD value.

* * * * *

US006526039B1

(12) **United States Patent**
Dahlman et al.

(10) **Patent No.: US 6,526,039 B1**
(45) **Date of Patent: Feb. 25, 2003**

(54) **METHOD AND SYSTEM FOR FACILITATING TIMING OF BASE STATIONS IN AN ASYNCHRONOUS CDMA MOBILE COMMUNICATIONS SYSTEM**

(75) Inventors: **Erik Bengt Lennart Dahlman**, Bromma (SE); **Karim Jamal**, Tokyo (JP); **Johan Anders Nyström**, Stockholm (SE); **Mats Cedervall**, Vallentuna (SE); **Patrik Lundqvist**, Stockholm (SE)

(73) Assignee: **Telefonaktiebolaget LM Ericsson**, Stockholm (SE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/243,095**

(22) Filed: **Feb. 2, 1999**

Related U.S. Application Data

(60) Provisional application No. 60/074,494, filed on Feb. 12, 1998.

(51) **Int. Cl.**[7] **H04B 7/216**; H04B 1/713

(52) **U.S. Cl.** **370/350**; 370/342; 375/150; 375/152; 375/142; 375/143; 455/456

(58) **Field of Search** 370/342, 335, 370/350; 375/145, 149, 140, 142, 143, 180, 152; 455/501, 502, 500, 442, 67.6, 524, 456

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,440,561 A		8/1995	Werronen	370/337
5,828,659 A	*	10/1998	Teder et al.	370/342

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0 241 565	10/1987	H04Q/7/04
WO	94/30024	12/1994	H04Q/7/04
WO	97/47148	12/1997	H04Q/7/32

OTHER PUBLICATIONS

TIA-EIA Interim Standard, *Mobile Station—Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System,* TIA/EIA/IS-95-A, May 1995.

A. Baier, et al., "Design Study for a CDMA-Based Third-Generation Mobile Radio System," *IEEE JSAC,* vol. 12, pp. 733–743, May 1994.

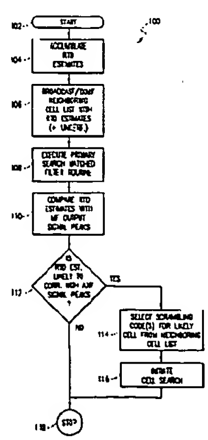
(List continued on next page.)

Primary Examiner—William Trost
Assistant Examiner—Rafael Perez-Gutierrez
(74) *Attorney, Agent, or Firm*—Jenkens & Gilchrist, PC

(57) **ABSTRACT**

A method and system are disclosed for facilitating the timing (e.g., the known relative timing differences) of base stations (BSs) in asynchronous CDMA mobile communications systems. A plurality of mobile stations (MSs) measure the relative time differences between various pairs of BSs, and these measurements are stored by the BSs. A source BS sends to an MS, in a neighbor list message, estimates of the relative time difference between the source BS and each of the BSs on the neighboring cell list. Each BS on the list can maintain a relative time difference estimate table, which can be updated continuously from the reports received from MSs. Subsequently, the BSs can send entries from this table to the MS in the neighbor list message. Using this novel technique, the BSs have known relative timing differences. Consequently, when the MS initiates a cell-search for a candidate BS, the MS already has an estimate of the timing of that BS as compared to its source BS. As such, the resulting cell-search procedure has a lower level of complexity and thus can be accomplished much quicker than with prior procedures. In addition, the relative time difference estimates can be compared with corresponding time differences that are measured by a second mobile station. Based on this comparison, the propagation delays of signals between the second MS and various BSs can be calculated to determine the position of the second MS.

42 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

5,859,612	A	* 1/1999	Gilhousen	342/457
5,872,774	A	* 2/1999	Wheatley, III et al.	370/335
5,991,330	A	* 11/1999	Dahlman et al.	370/509
6,072,847	A	* 6/2000	Dupuy et al.	375/356
6,097,709	A	* 8/2000	Kuwabara	370/350
6,101,175	A	* 8/2000	Schorman et al.	370/350
6,151,311	A	* 11/2000	Wheatley, III et al.	370/335
6,208,871	B1	* 3/2001	Hall et al.	370/335
6,246,673	B1	* 6/2001	Tiedemann, Jr. et al.	370/335
6,259,683	B1	* 7/2001	Sekine et al.	370/350
6,307,840	B1	* 10/2001	Wheatley, III et al.	370/335
2001/0021179	A1	* 9/2001	Tiedmann, Jr. et al.	370/335
2001/0022779	A1	* 9/2001	Wheatley, III et al.	370/350
2001/0046240	A1	* 11/2001	Longoni et al.	370/328

OTHER PUBLICATIONS

"Concept Group Alpha—Wideband Direct–Sequence CDMA (WCDMA)," *Evaluation Document* (in four parts), ETSI SMG Meeting No. 24, Madrid, Dec. 15–19, 1997.

PCT International Search Report, mailed May 18, 1999.

* cited by examiner

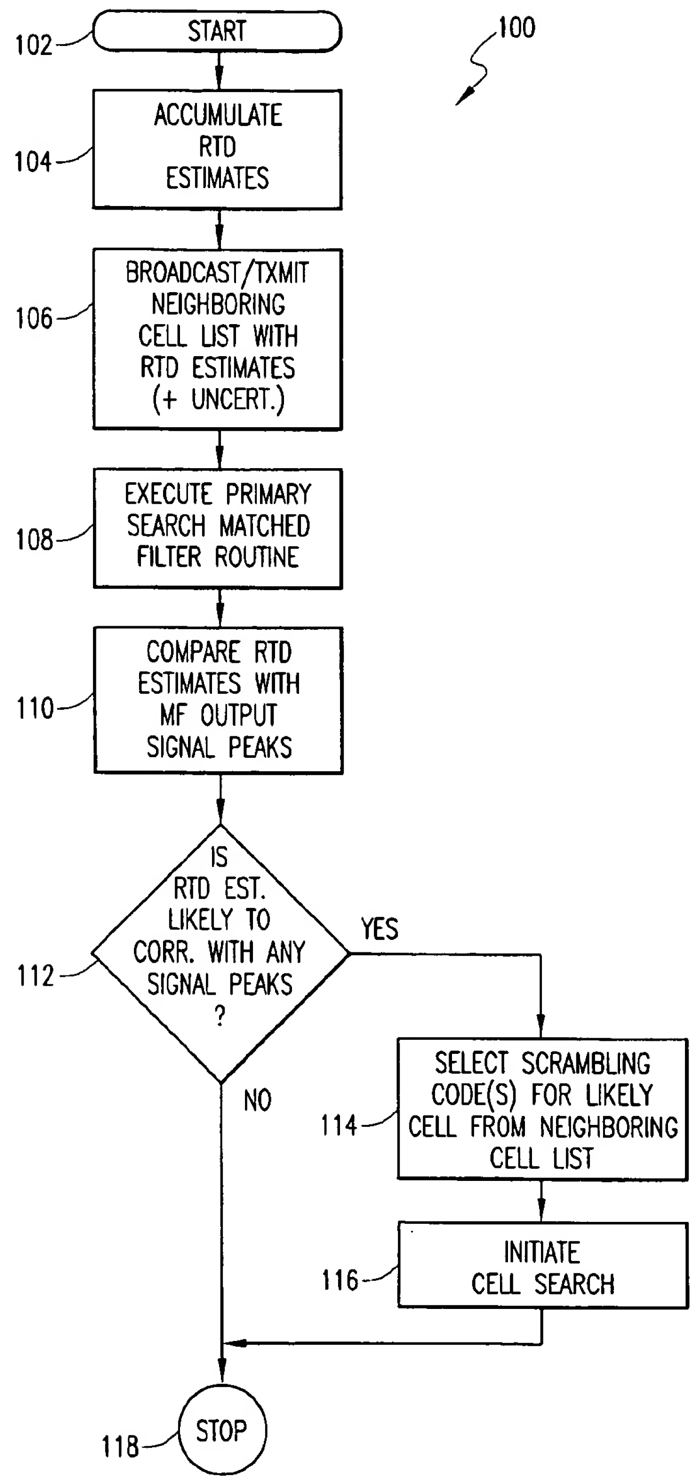

100
102
START
104
ACCUMULATE RTD ESTIMATES
106
BROADCAST/TXMIT NEIGHBORING CELL LIST WITH RTD ESTIMATES (+ UNCERT.)
108
EXECUTE PRIMARY SEARCH MATCHED FILTER ROUTINE
110
COMPARE RTD ESTIMATES WITH MF OUTPUT SIGNAL PEAKS
112
IS RTD EST. LIKELY TO CORR. WITH ANY SIGNAL PEAKS ?
YES
NO
114
SELECT SCRAMBLING CODE(S) FOR LIKELY CELL FROM NEIGHBORING CELL LIST
116
INITIATE CELL SEARCH
118
STOP

FIG. 1

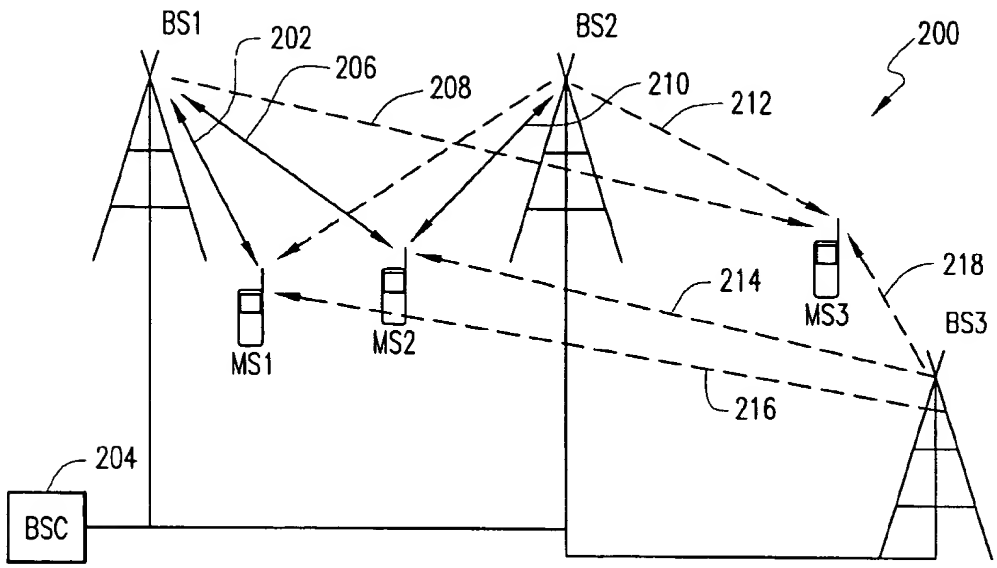

BS1
202
206
208
BS2
210
212
200
218
214
MS3
BS3
MS1
MS2
216
204
BSC

FIG. 2

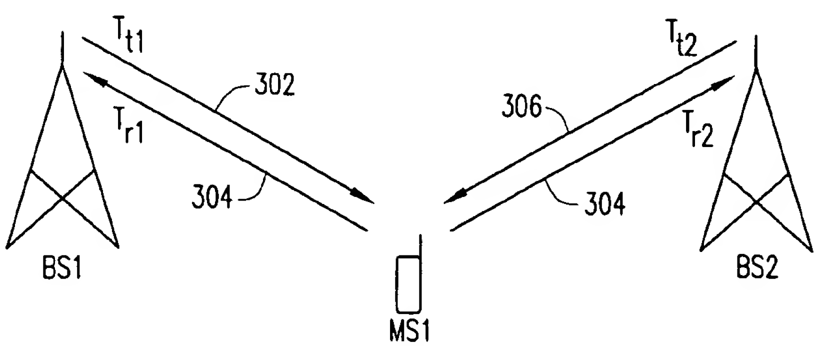
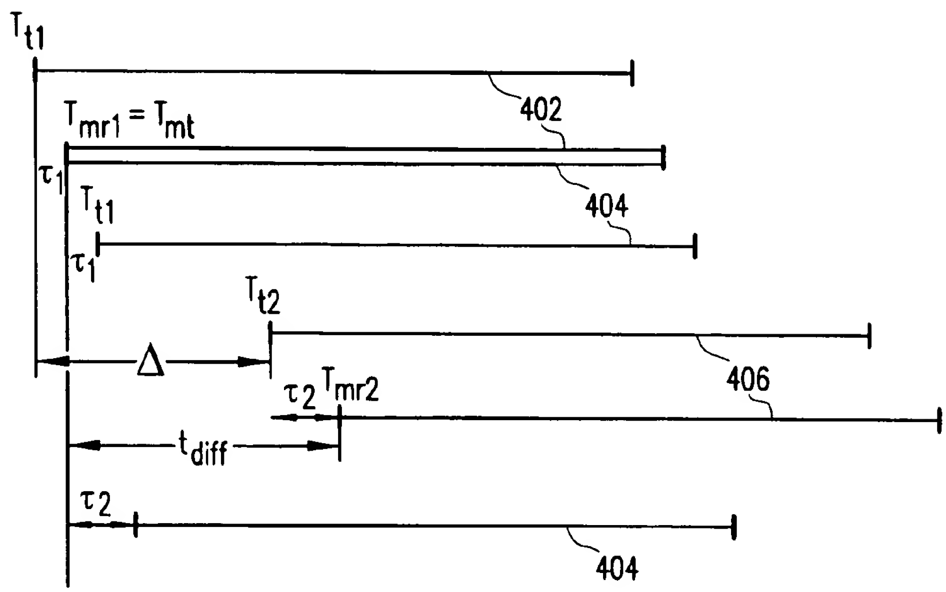

T_{t1}
302
T_{r1}
304
T_{t2}
306
T_{r2}
304
BS1
MS1
BS2

FIG. 3



T_{t1}
402
$T_{mr1} = T_{mt}$
τ_1
404
T_{t1}
τ_1
T_{t2}
Δ
406
τ_2 T_{mr2}
t_{diff}
τ_2
404

FIG. 4

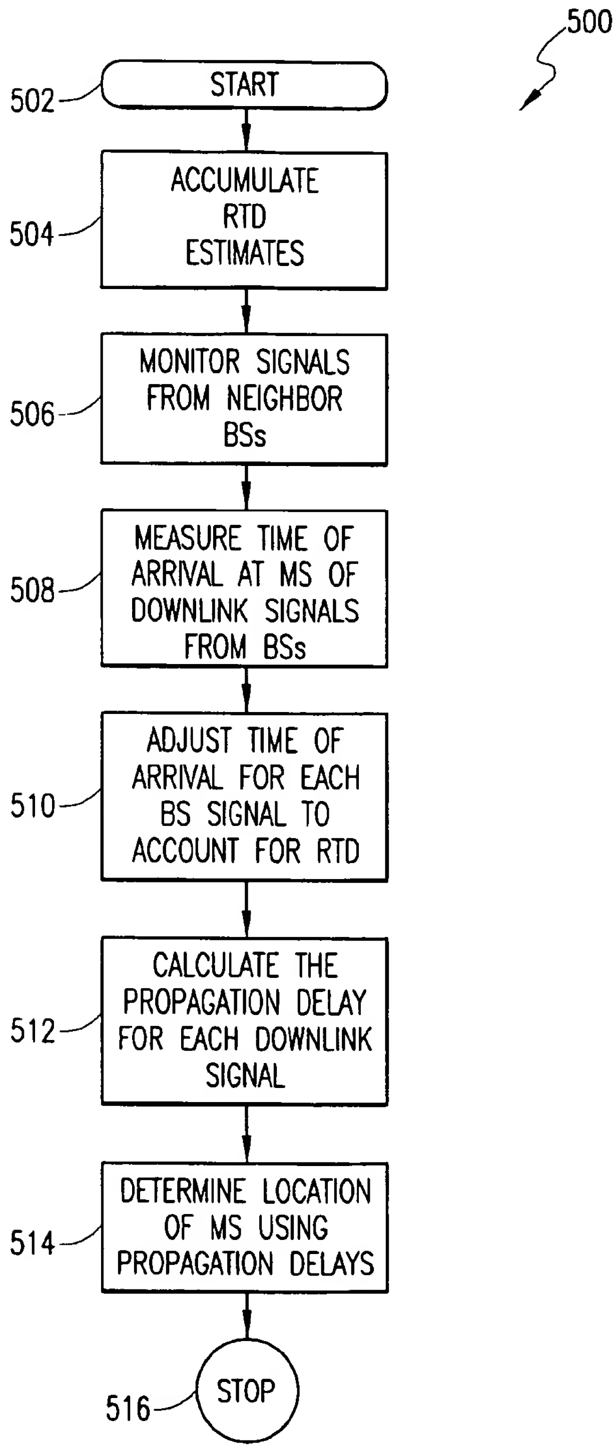

500
502
START
504
ACCUMULATE RTD ESTIMATES
506
MONITOR SIGNALS FROM NEIGHBOR BSs
508
MEASURE TIME OF ARRIVAL AT MS OF DOWNLINK SIGNALS FROM BSs
510
ADJUST TIME OF ARRIVAL FOR EACH BS SIGNAL TO ACCOUNT FOR RTD
512
CALCULATE THE PROPAGATION DELAY FOR EACH DOWNLINK SIGNAL
514
DETERMINE LOCATION OF MS USING PROPAGATION DELAYS
516
STOP

FIG. 5

METHOD AND SYSTEM FOR FACILITATING TIMING OF BASE STATIONS IN AN ASYNCHRONOUS CDMA MOBILE COMMUNICATIONS SYSTEM

RELATED APPLICATION

This application claims the benefit of the filing date of U.S. Provisional Application No. 60/074,494, filed Feb. 12, 1998.

BACKGROUND OF THE INVENTION

1. Technical Field of the Invention

The present invention relates in general to the mobile communications field and, in particular, to a method and system for facilitating the timing of base stations in an asynchronous Code-Division Multiple Access (CDMA) mobile communications system.

2. Description of Related Art

Direct-Sequence CDMA (DS-CDMA) mobile communications systems can be either inter-cell synchronous or inter-cell asynchronous systems. In other words, the base stations (BSs) in an inter-cell synchronous system are accurately synchronized with one another, and the BSs in an inter-cell asynchronous system are not. More specifically, asynchronous BSs do not share a common time reference, and their transmissions, therefore, have arbitrary, not predetermined timing relative to each other. An example of an inter-cell synchronous system is the North American IS-95 system. Examples of inter-cell asynchronous systems are the Wideband CDMA (WCDMA) systems proposed in the CODIT, ETSI SMG2 Group Alpha, and ARIB technical specifications.

The main disadvantage of inter-cell synchronous systems is that the BSs have to be very accurately synchronized (down to the μs level). This high level of accuracy is typically provided through the use of highly accurate time references co-located with the BSs, such as Global Positioning System (GPS) receivers. However, because of the line-of-sight nature of satellite signal propagation, the use of such co-located references are likely not feasible for BSs located underground, in buildings or tunnels. Another related disadvantage is that the GPS system is controlled by a government agency. Consequently, the use of GPS receivers for BS network synchronization may be undesirable in some national regions. These disadvantages are the main reasons why inter-cell asynchronous systems are now being considered.

For inter-cell asynchronous systems to work properly, there are two crucial functional issues that need to be addressed: (1) Soft Handovers (SOHOs); and (2) Cell-Searches. In a state of SOHO, a mobile station (MS) is in communication with more than one BS at the same time. To facilitate the SOHOS, the MS constantly scans for other BSs in the vicinity. The MS can thereby monitor the received signal quality from the multiple BSs and determine the time delay of the BSs. For a SOHO to occur, the MS being handed over has to be able to receive the "target" BS's signal at approximately the same time as the "source" BS's signal, in order to minimize buffering requirements (i.e., a smaller time difference between BS signals requires less buffer area than larger time differences). Also, the target BS has to be able to find the MS's signal without an unreasonable expenditure of processing resources.

These SOHO issues are resolved for asynchronous systems by a "per-call" synchronization technique, which is

disclosed in "A Design Study for a CDMA-Based Third-Generation Mobile Radio System," by A. Baier et al., *IEEE JSAC*, Vol. 12, pp. 733–743, May 1994. Using this technique, the MS involved in the SOHO calculates and reports to the network the time difference between the target BS and source BS. The network notifies the target BS via the Base Station Controller (BSC) or Radio Network Controller (RNC) about the time difference. The target BS can then adjust its receive and transmit timing for the signal intended for the MS involved, to compensate for the difference.

A similar known SOHO technique is used in which the MS reports the timing difference between the target BS's transmission and its own transmission, rather than the difference between the target BS's transmission and the source BS's transmission. However, since the MS's transmit/receive timing relationship is always fixed, the two above-described SOHO techniques are essentially equivalent. These techniques are referred to as mobile assisted handover (MAHO). In other words, the MS assists the target BS in compensating for the difference in timing between the target BS and source BS.

A cell-search generally refers to a procedure whereby an MS accomplishes chip-, slot- and frame-synchronization with a BS, and detects the BS's downlink scrambling code. This procedure is used both during power on (initial synchronization) and continuously thereafter during the idle or active modes while the MS is searching for SOHO candidate BSs. In a synchronous system, the cell-search can be performed efficiently (i.e., with a relatively low level of complexity) because the same scrambling code can be used by all BSs. As such, the MS can perform the complete search for BSs using only a single matched filter (or a similar functionality). However, this same technique cannot be readily used in an asynchronous system because of the different scrambling codes used by the different BSs. Consequently, a need has arisen for a low-complexity, rapid cell-search procedure for asynchronous CDMA systems.

A rapid, multi-step cell-search procedure for asynchronous CDMA systems has been proposed, whereby each BS transmits one unmodulated symbol. This transmitted symbol is spread by a globally-known short code, without a scrambling code, in each slot of each frame. In one such proposal, this symbol is denoted as a "Perch 1 Long Code Masked Symbol (LCMS)". In a second proposal, this symbol is denoted as a "Primary Synchronization Channel" or Primary (SCH). With the proposed multi-step procedure, an MS can thus find the chip- and slot-timing of a BS, using a single matched filter which is matched to the Primary SCH. Subsequently, the MS still has to find the BS's frame-timing and downlink scrambling code (which spans one frame in the proposed multi-step procedure). The MS can find the BS's frame-timing by detecting a second regularly transmitted symbol, which is denoted as a "Perch 2 LCMS" or "Secondary SCH".

This second symbol is transmitted in parallel with the first symbol, but the second symbol is spread by a second short code (again without a scrambling code). The second symbol may also have a unique repetitive modulation pattern per frame, and by detecting this pattern, the MS can determine the BS's frame-timing. The spreading code used for the second symbol indicates to the MS which group of possible scrambling codes an actually-used scrambling code belongs to. The MS can then find the scrambling code used, by correlating with the scrambling codes belonging to the indicated group, at the above-identified frame-timing (or at different possible frame-timings). However, a problem with the proposed multi-step procedure is that the level of com-

plexity of the cell-search is still relatively high, especially in the case of a SOHO candidate search (which the MS has to perform on a regular basis).

Another problem with inter-cell asynchronous systems is that the timing difference between BSs makes it difficult to determine the position of the MSs. Mobile communications systems capable of determining the position of MSs in the system are becoming increasingly desirable. Currently, mobile positioning is generally performed by the use of external systems, such as a GPS system. Preferably, however, mobile positioning would be performed by the cellular system itself without the need for such external systems. To perform such cellular positioning, a method is needed to accurately determine the absolute or relative distances between an MS and each of several different BSs. The distances can be calculated using propagation time, time of arrival (TO), or time difference of arrival (TDOA) measurements on the signals transmitted between the MSs and each of several different BSs. Once these measurements are available, a number of algorithms exist to calculate the geographical position of the MS. For example, according to the TOA method, the distance from an MS to each of the BSs is obtained using TOA measurements. Each of these distances can be conceptualized as the radius of a circle with the respective BS in the center. In other words, the TOA measurement can be used to determine the radial distance of the MS from a particular BS, but the direction cannot be determined based on a single TOA measurement; thus, the MS might lie anywhere on the circle defined by the calculated radius. By determining the intersection of the circles associated with each of several different BSs, however, the position of the MS can be determined. The TDOA method, on the other hand, uses the difference in TOA between two BSs to determine a TDOA between those two BSs. The position of the MS can then be estimated to be along a curve, namely a hyperbola, in accordance with the TDOA calculation. By using three or more BSs, more than one such curve can be obtained. The intersection of these curves gives the approximate position of the MS.

In the simplest mobile positioning technique, a SOHO is made to a number of BSs. During each of these handovers, the propagation time between each BS and the MS can be measured. The location of the MS can then be determined by triangulating the position of the mobile. This positioning method is the simplest to implement because it involves very little change in the mobile radio design. In addition, the BSs do not need an absolute time reference; i.e., this method may be used in an asynchronous cellular system. However, because of the geographical separation between BSs, handover to two other geographically located BSs is only possible in a small number of cases. In other words, when the MS is in close proximity to one BS, a SOHO with other BSs will often not be possible. This is because the "hearability" of signals between the MS and multiple BSs will normally be unsatisfactory.

Another possible solution is to use an antenna array at the BS. When the BS has an antenna array, the position of the MS can be calculated by estimating the direction from which uplink signals are propagating and by measuring the round-trip delay of the communications signal. In this method, the MS only needs to be in communication with one BS to calculate the position. However, widespread use of antenna arrays for positioning purposes is expensive. Furthermore, the effects of multipath propagation characteristics of the uplink and downlink signals often make an antenna array undesirable, particularly in cities, where signals frequently reflect off buildings and other structures.

As mentioned above, it is also possible that a GPS can be incorporated into the mobile without using an extra radio receiver. This method, however, requires excessive computational and receiver complexity in the MS.

Another solution is to measure the propagation time, TOA, or TDOA of signals transmitted by the BSs to the MS or by the MS to the BSs. For example, a downlink solution can be used wherein, in the case of CDMA, the MS measures the TOA of pilot channel data that is transmitted by several different BSs. Alternatively, an uplink solution can be used wherein several BSs each measure the TOA of a signal transmitted by the mobile to the multiple BSs. However, both of these methods require an absolute or accurate relative time reference in, or synchronization of, the BSs. Therefore, both downlink and uplink solutions normally require extra hardware (e.g., a GPS receiver located in the BSs to obtain timing of the BSs) in an asynchronous network.

A system and method are needed for reducing the complexity of and the processing resources used during the cell search and mobile positioning processes in asynchronous networks. In particular, it would be advantageous to utilize as much a priori search information as possible to help reduce the level of complexity and increase the search rate for cell-searches and to enable simplified mobile positioning solutions. As described in detail below, the present invention successfully resolves the above-described problems.

SUMMARY OF THE INVENTION

A method and system are provided for facilitating the timing of base stations in asynchronous CDMA mobile communications systems, whereby a source BSC (or RNC) sends to an MS (e.g., in a neighbor cell list message) estimates of the Relative Time Difference (RTD) between the source BS and each of the BSs on the neighboring cell list. For SOHO purposes, a plurality of MSs can report to the network the estimated RTDs along with signal quality information for the neighboring BSs. Each BS can maintain an RTD estimate table, which can be updated continuously from the RTD reports received from the MSs. Subsequently, the BSs can send entries from this RTD estimate table to the MS in the neighboring cell list message, along with corresponding scrambling codes. Using this novel technique, the BSs have known relative timing differences. Consequently, when the MS initiates a cell-search for a potential target BS, the MS already has an estimate of the timing of that BS as compared to its source BS. As such, the resulting cell-search procedure used in an asynchronous CDMA system has a lower level of complexity and thus can be accomplished much quicker than with prior procedures.

In another aspect of the invention, the accuracy of the estimated RTD's can be greatly improved by accounting for propagation delays between the MS and the BSs that are used to estimate the RTD. These improved RTDs can be used to further improve timing estimates for performing cell-searches. The improved RTDs can also be used to calculate the position of MSs in the mobile communications system. Once highly accurate RTDs are known, distances between an MS and several BSs can easily be determined using the propagation times, TOAs, or TDOAs of signals traveling between the MS and the several BSs.

An important technical advantage of the present invention is that neighboring BSs in an asynchronous CDMA mobile communications system have known relative timing differences.

Another important technical advantage of the present invention is that the hardware and software complexity of

MSs in an asynchronous CDMA mobile communications system is reduced.

Yet another important technical advantage of the present invention is that the overall level of complexity of the cell-search procedure in an asynchronous CDMA mobile communications system is significantly reduced.

Still another important technical advantage of the present invention is that the speed of the cell-searches performed in asynchronous CDMA mobile communications systems is significantly increased as compared to prior procedures.

Another important technical advantage of the present invention is that mobile positioning can be determined in an asynchronous mobile communications system by performing simple calculations on easily obtainable data and without the need for an external system.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and apparatus of the present invention may be had by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

FIG. **1** is a flow diagram that illustrates an exemplary method that can be used for facilitating the timing of base stations in an asynchronous CDMA mobile communications system, in accordance with a preferred embodiment of the present invention;

FIG. **2** is a simplified block schematic diagram of an exemplary mobile communications system that can be used to implement the method shown in FIG. 1, in accordance with the preferred embodiment of the present invention.

FIG. **3** is a simplified block schematic diagram of an MS that is in or is about to enter a SOHO and that can be used for facilitating improved timing calculations of BSs in an asynchronous CDMA mobile communications system, in accordance with a preferred embodiment of the present invention;

FIG. **4** is a diagram of the relative timing of signals involved in the SOHO scenario depicted in FIG. 3; and

FIG. **5** is a flow diagram that illustrates an exemplary method that can be used to determine the position of an MS, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE DRAWINGS

The preferred embodiment of the present invention and its advantages are best understood by referring to FIGS. **1–5** of the drawings, like numerals being used for like and corresponding parts of the various drawings.

Essentially, in an asynchronous CDMA system, a BSC "knows" the downlink scrambling codes for all of its BSs. Typically, a list of neighboring cells is broadcast in each cell (for MSs operating in the idle mode), or transmitted on a dedicated control channel (for MSs operating in the active mode). When an MS receives the neighboring cell information, it determines the scrambling codes of the listed neighboring cells that are potential SOHO candidate cells. Having such a priori knowledge of this scrambling code information for the candidate SOHO cells enables the MS to reduce the total SOHO cell-search time (or complexity level), because the number of possible scrambling codes is reduced in comparison with the number for initial synchronization (power on). However, even if the set of scrambling codes to be searched by the MS is relatively small, the MS still does not know the timing of these codes. This lack of timing information is the main reason why current proposals for an asynchronous system cell-search take more time (and are more complex) than a synchronous system cell-search.

The present invention solves this lack of timing information problem by having the source BS send to the MS (along with the neighboring cell list) an estimated RTD between the source BS and each of the BSs on the neighboring cell list. In other words, instead of sending only the scrambling codes of the neighboring BSs to the MS, the source BS also transmits each of their estimated RTDS. For SOHO purposes, the MSs can report (on a regular basis, triggered by some event, or on demand from the BSC) to the network the estimated RTDs along with signal quality information (e.g., signal strength, signal-to-interference ratio or SIR, etc.) for the neighboring BSs. Consequently, each BSC can maintain an RTD estimate table, which can be updated continuously from the RTD reports received from the MSs. In a preferred embodiment of the present invention, the RTD estimate table is maintained in a database at the BSC.

Subsequently, the BSCs can send entries from this RTD estimate table to the MS in the neighboring cell list message, along with the corresponding scrambling codes (with the BSC keeping track of the estimated RTD information it has already sent in previous messages to the MS). Using this novel technique, the BSs have known relative timing differences. Consequently, in an exemplary embodiment, when an MS initiates a search for a potential target BS, the MS already has an estimate of the timing of that BS (i.e., from the RTD information) as compared to its source BS. As such, the resulting cell-search procedure used in an asynchronous CDMA system can be accomplished much quicker than with prior procedures. When the MS has synchronized with the potential target BS, the MS has an improved estimate of the RTD, which in turn, the MS can report back to the source BS (preferably along with quality information for the potential target BS). The source BS (or its associated BSC) can then update this entry in the RTD estimate table.

More specifically, FIG. 1 is a flow diagram that illustrates an exemplary method **100** that can be used for facilitating the timing of BSs and increasing the speed of hand-over candidate cell-searches in an asynchronous CDMA mobile communications system, in accordance with a preferred embodiment of the present invention. At step **104** of the exemplary method shown in FIG. 1, a BSC prepares a neighbor cell list (e.g., "neighbor set" in an IS-95 system) with respective scrambling codes, along with a plurality of RTD estimates between a source BS and the respective hand-over candidate BSs from an RTD estimate table (preferably maintained in a database at the BSC). At step **106**, the source BS broadcasts or transmits the neighbor cell list with scrambling codes and RTD estimates in a "neighbor list message" to the MS involved. In actuality, the BSC keeps track of the estimated RTDs it sends to the MS, in order not to unnecessarily duplicate RTD information the MS may already have. At this point, the MS has now received a list of the BSs it can synchronize with (and also report quality information for). The received neighbor list message can also include an uncertainty estimate (described in more detail below). The MS stores the neighbor cell list information in local memory.

At step **108**, with the a priori neighbor cell RTD estimate (timing) information readily at hand, along with the other corresponding neighbor cell information, the MS can initiate a primary cell-search using a conventional matched filter arrangement. The MS's utilization of the primary cell-search matched filter produces signal peaks that correspond to the BSs that the MS can receive with sufficient quality to qualify as hand-over candidate cells. At step **110**, the MS correlates

the RTD estimates with the produced matched filter signal peaks to determine which peaks are most likely to correspond to which scrambling codes in the neighbor cell list (step 112). At step 114, based on the correlations produced at step 112, the MS can select the scrambling codes for the most likely hand-over candidate cells from the neighbor cell list. The MS can then initiate the cell-search (step 116).

Theoretically, if the above-described RTD estimates are perfectly accurate, then the MS could (up-front) discard all of the matched filter output signal peaks not corresponding to the RTD "estimate" information. In this hypothetical situation, the scrambling code correlation procedure (e.g., step 112) could be omitted altogether. However, in any event, in accordance with the present invention, the MS's utilization of the RTD estimates to determine the most likely hand-over candidate cells from the neighbor cell list enables the MS to disregard a significant number of the matched filter peaks, and/or associate certain of those peaks with corresponding scrambling codes, which significantly reduces the complexity of the cell-search procedure and substantially increases the speed of the search.

By using the above-described inventive method, each BS (cell), with the assistance of the MSs connected to it, has a known relative timing difference with respect to its neighbor BSs (cells). If, for some reason, there are no MSs connected to a particular BS, the RTD estimate table corresponding to that BS is not updated. Consequently, since the relative timing between the neighboring BSs may be continually shifting, the uncertainty (or variance) of the RTD estimate table entries for this BS will increase. In general, the uncertainty of the RTD estimate may increase with time, but this uncertainty is typically minimal immediately after an update has been completed (e.g., based on an RTD report received from an MS). Consequently, in order for the communications system to be more robust during periods of MS inactivity (e.g., at night, or during holidays in private indoor systems), as mentioned earlier, an RTD uncertainty estimate can be broadcast or transmitted from the BS along with the RTD estimate, in the neighbor list message. The MS can then, for example, set (e.g., increase) its time-search window accordingly to allow for the additional level of uncertainty. The MS can thus cope with those BSs having a relatively uncertain knowledge of its RTDs, and also minimize its complexity level when relatively certain RTD estimates have been provided.

An additional method for further mitigating the uncertainty problem encountered when there are too few active MSs for relatively long periods is to place "dummy" MSs at fixed locations throughout the system. These "dummy" MSs can have a limited functionality, and can be called upon by BSs having relatively high uncertainty RTD estimate table entries to provide more current RTD updates. Such "dummy" MSs can be thus advantageously located where they can be reached by a plurality of BSs (e.g., near the cell borders).

FIG. 2 is a simplified block schematic diagram of an exemplary mobile communications system 200 that can be used to implement the method 100 (FIG. 1) for facilitating the timing (e.g., the known relative timing differences) of BSs and increasing the speed of cell-searches, in accordance with the preferred embodiment of the present invention. System 200 is preferably an asynchronous CDMA mobile communications system that includes, for illustrative purposes, three BSs and three MSs. However, it should be understood that the number of BSs and MSs shown is for illustrative purposes only, and that a typical system can include more than three BSs and three MSs. For this example, MS1 is operating in the active mode and connected via air interface link 202 to BS1. In accordance with step 106 of method 100 (FIG. 1), MS1 has received a neighbor list message preferably including respective RTD estimates and, optionally, associated uncertainty estimates on a dedicated control channel from the BSC 204 (via BS1 once it is "connected" to MS1). At least two of the neighbors (cells) listed as entries in the RTD estimate table are BS2 and BS3. On a periodic basis (or on demand), MS1 monitors and reports the quality (signal strength, SIR, signal-to-noise ratio or SNR, Bit-Error-Rate or BER, etc.) of those BSs to the BSC 204 (via BS1). Since MS1 has received RTD estimates from BSC 204 (via BS1), MS1 can synchronize itself relatively rapidly with BS2 and BS3, at least during the first occasion when MS1 searches for BS2 and BS3. When MS1 has synchronized with BS2 (or BS3), it can be assumed that MS1 has a "good" RTD estimate for that BS. On a periodic basis, or on demand, MS1 can report the estimated signal quality of at least one of the entries in the neighbor cell list to BSC 204 (via BS1). In addition to the quality estimates, MS1 can also report the current RTD estimate to BSC 204.

The cell-search situation for MS2 is similar to that of MS1, except for the example shown, MS2 is involved in a SOHO with both BS1 and BS2, and monitors only one other BS (e.g., BS3 via air interface link 214). For this example, MS3 is operating in the idle mode (has no connection set up), but it can still monitor the BSs according to the neighboring-cell list received on the broadcast channel of the BS the MS3 considered the "best one" to listen to (e.g., in this case BS3 via air interface link 218). As such, MS3 can also monitor BS1 (via air interface link 208) and BS2 (via air interface link 212). Again, the RTD estimates broadcast by BS3 assists MS3 in synchronizing more rapidly with BS1 and BS2, or at least the first time the synchronization procedure occurs. The complexity of the cell-searches are thus reduced, and the speed of the cell-searches is thereby significantly increased.

Preferably, each MS operating in the mobile communications system 200 will transmit its measured RTD estimate on a periodic basis, or on demand, to the BSC 204 (via BS1). The BSC 204 stores the RTD estimates received from the MSs in an RTD estimate table. Alternatively, each entry stored in the RTD estimate table (i.e., representing an estimated difference between a pair of base stations) can be calculated based on estimates received from a plurality of different MSs. For example, the stored estimate can constitute an average of the previous x received estimates, or of the estimates received in the preceding y minutes. The values in the table can be updated by replacing previous estimates or by recalculating particular estimates based on newly received data. The values stored in the table are then sent to other MSs, as described above, along with the neighbor cell list, to assist in synchronizing those MSs with neighboring BSs, as necessary. In addition, it will be appreciated by those skilled in the art that the RTD estimate table does not have to be stored in the BSC 204; rather, the table can be stored in one or more databases located virtually anywhere in the network (e.g., in a register associated with the MSC or in an entirely separate database).

In another aspect of the invention, the RTD estimates can be used to determine the position of the MS. Positioning calculations, however, require more accurate RTD estimates than in the case of cell searches. This is because the mobile positioning concept essentially relies upon a determination of the propagation delay between the MS and each of a plurality of BSs or upon TOA or TDOA measurements among the various BSs. In most cases, the speed of the cell

search can be significantly improved without having to account for propagation delays. Thus, it is normally sufficient to base the RTD estimates on the time difference between two BSs as measured by one or more mobiles without considering the effect of propagation delays of the downlink signals received by an MS from each of the BSs. To perform mobile positioning, on the other hand, a more accurate estimate of the RTD is needed.

The present invention solves this problem by calculating an improved RTD that accounts for the propagation delays of uplink and downlink signals. Essentially, the improved RTD is the difference between the time at which a first BS begins transmitting its downlink signal and the time at which a second BS begins transmitting its downlink signal. This improved RTD estimate can be calculated using: (1) the local receive and transmit times of the uplink and downlink signals in the BSs of interest, as measured at each of the respective BSs, and (2) the TOA difference at the MS of the downlink signals from the BSs, as reported by the mobile. This improved RTD information can then be used by other mobiles for positioning purposes.

In a preferred embodiment, the improved RTD estimates are stored in a database table at the BSC or the MSC. Subsequently, a position determination for a second MS is desired (on a regular basis, triggered by some event, or upon request by the BSC or the MS). The second MS measures the time differences between the BSs based on the receive time at the second MS of the downlink signal from each BS and reports the measured time differences to the BSC. The BSC then compares the stored improved RTD estimate between a particular pair of BSs with the measured time difference between the same pair of BSs as reported by the MS. Based on this comparison, the propagation delays between each of several BSs and the MS can be calculated, and an accurate determination of the MS's location can be made. Again, TOA or TDOA measurements can also be used to determine the location. Regardless of which positioning method is used, however, the positioning calculations essentially rely upon the existence of propagation delays in the mobile environment.

Generally, the determination of each RTD estimate by an MS involves only two BSs, even if a three-part SOHO (i.e., a SOHO involving three different BSs) can occur. By repeating the RTD determination during multiple different SOHO procedures, an improved, estimated RTD between a substantial number of possible pairs of BSs can be determined. The improved RTD estimates are normally then used by other MSs (i.e., MSs that were not involved in the estimated RTD calculations) to determine the position of those other MSs. it will be appreciated, however, that the position of an MS that was involved in the estimated RTD calculations can also be determined using the improved RTD estimates. In any event, the positioning procedure preferably utilizes as many BSs as possible in order to improve the accuracy of the estimated location.

Referring now to FIG. 3, there is shown a schematic illustration of an MS that is in or is about to enter a SOHO. A first base station BS1 transmits a frame of a downlink signal **302** (either a pilot frame or a traffic data frame) at a time T_{t1}, as measured in the time base of the first base station BS1. An uplink signal **304** from a mobile station MS1 is received by the first base station BS1 at a time T_{r1}, also measured in the time base of the first base station BS1. Similarly, a second base station BS2 transmits a downlink signal **306** at a time T_{t2} and receives the uplink signal **304** from the mobile station MS1 at a time T_{r2}, measured in the time base of the second base station BS2. Generally, the time base of the two base stations in an asynchronous network will have a relative time difference (RTD) Δ. In other words, if an event (such as the transmission of a pilot frame) occurs at a time T_1 in the first base station BS1, a corresponding event will occur at a time

$$T_2=T_1+\Delta \tag{1}$$

in the second base station BS2. Once the RTD Δ is known, it can be used by other MSs for mobile positioning.

In addition, each downlink signal has an offset t_i relative to the transmit time of the pilot channel frame. Thus, the traffic channel data from the first base station BS1 is transmitted at a time

$$T_{t1}=T_{p1}+t_1, \tag{2}$$

where T_{p1} is the transmit time of the pilot channel frame from the first base station BS1. Similarly, the traffic channel data of the downlink signal from BS2 is transmitted at time

$$T_{t2}=T_{p2}+t_2. \tag{3}$$

When the SOHO is initialized, the mobile station MS1 simply listens to the pilot and t_2=0. Later, when the mobile station MS1 is in SOHO, the second base station BS2 will transmit data and the offset t_2 of the signals will be adjusted so that the data from the first base station BS1 and the second base station BS2 will arrive at the mobile at approximately the same time. In the following discussion, a generic scenario can be considered in which it is assumed that the offsets t_1 and t_2 are known. This scenario covers both the cases of SOHO initialization and an already set-up SOHO.

Referring now to FIG. 4, there is illustrated a relative timing diagram of the various SOHO signals that are transmitted and received in the system of FIG. 3. All of the times in the figure are illustrated in a common, arbitrary time base. For purposes of making RTD calculations in accordance with the present invention, however, the time of each event is reported in the local time base of the station (i.e., the MS or the BS) associated with that event.

At a time T_{t1}, as measured in the time base of the first base station BS1, a pilot or traffic frame **402** is transmitted by the first base station BS1. The frame **402** is received at the mobile station MS1 at a time T_{mr1}, which is measured in the time base of the mobile station MS1. The time T_{mr1} is delayed after the transmission time T_{t1} by a propagation delay time τ_1, which is the time required for the signal to travel from the first base station BS1 to the mobile station MS1 and vice versa. The mobile station MS1 transmits its uplink signal **404** at time T_{mt}. For simplicity and without loss of general applicability, it can be assumed that the mobile station MS1 transmits its uplink signal **404** at the same time it receives the downlink signal **402** from the first base station BS1. Thus,

$$T_{mt}=T_{mr1}, \text{ and} \tag{4}$$

the uplink signal **404** is received in the first base station BS1 at time

$$T_{r1}=T_{t1}+2\tau_1. \tag{5}$$

The uplink signal **404** from the mobile station MS1 is received in the second base station BS2 at a time T_{r2} and is delayed after the transmission time T_{mt} by a propagation delay time τ_2.

The second base station BS2 also transmits a traffic or pilot frame **406** at a time T_{t2}. After the propagation delay

time T_2, the downlink signal **406** is received by the mobile station MS1 at a time T_{mr2}.

To calculate the RTD Δ, the mobile station MS1 reports the time difference t_{diff} between the reception time T_{mr2} of the downlink signal **406** from the second base station BS2 and the transmit time T_{mt} of the uplink signal **404** from the mobile station MS1. Thus,

$$t_{diff}=T_{mt}-T_{mr2}. \tag{6}$$

It should be noted that in FIG. **4**, the time difference t_{diff} is relatively large, which is typically indicative of an initial acquisition scenario.

Using the above notations, we can then formulate the following expression for the receive time T_{r2} of the uplink signal in the second base station BS2:

$$T_{r2}=2\tau_2+t_{diff}+T_{t2}. \tag{7}$$

Finally, we can formulate the following expression for t_{diff}:

$$t_{diff}=T_{t1}-T_{t2}+\tau_1-\tau_2+\Delta, \tag{8}$$

which is obtained by subtracting the arrival time T_{mr2} of the downlink signal **406** from the second base station BS2 (either the pilot frame at SOHO initialization or the traffic data during SOHO) from the transmission time T_{mt} of the uplink signal **404** from the mobile station MS1, all measured in the time base of the second base station BS2. Thus, as will be understood by persons of ordinary skill in the art, in the time base of the second base station:

$$T_{mr2}=T_{t2}-t_2, \text{ and} \tag{9}$$

$$T_{mt}=T_{t1}+\Delta+\tau_1. \tag{10}$$

Now there are three equations: (5), (7), and (8), and three unknowns: 1) the propagation delay time τ_1 between the mobile station MS1 and the first base station BS1; 2) the propagation delay time τ_2 between the mobile station MS1 and the second base station BS2; and 3) the time difference Δ between the first base station BS1 and the second base station BS2. It is easy to solve for Δ to get

$$\Delta=\tfrac{1}{2}(t_{diff}-T_{t1}-T_{r1}+T_{t2}+T_{r2}), \tag{11}$$

which provides a solution for the desired RTD Δ between the base stations BS1 & BS2.

According to a preferred embodiment of the invention, the mobile station MS1 reports the time difference t_{diff} and each of the base stations BS1 & BS2 report their respective transmit and receive times to the network. The computation of the RTD Δ is then made in the BSC or the MSC. In the alternative, the computation can be performed in the mobile station MS1 or in a base station once the necessary timing data is provided.

By calculating improved RTD estimates between various pairs of BSs in an asynchronous mobile communications system, an uplink solution or a downlink solution can be used to determine the position of MSs in the system without the need for an absolute time reference. For example, FIG. **5** is a flow diagram that illustrates one possible method **500** for facilitating the timing of BSs and determining the position of a selected MS in an asynchronous CDMA mobile communications system, in accordance with one embodiment of the present invention. As will be appreciated by those of ordinary skill in the art, numerous other positioning methods, such as TOA or TDOA, can also be used in connection with the improved RTD estimate to facilitate positioning in accordance with the invention. At step **504**, a BSC calculates a plurality of improved RTD estimates between various pairs of BSs that are controlled by the BSC or that are listed in the neighbor cell list. This calculation is made by using data provided by other MSs in the mobile communications system. Accordingly, the effect of propagation delays are taken into account so as to calculate highly accurate RTD estimates. Preferably, a table of these improved RTD estimates is maintained in a database at the BSC. At step **506**, the selected MS monitors the BSs in neighboring cells. For purposes of the present positioning method **500**, this involves monitoring, for example, a known sequence that is periodically transmitted by the BSs. This monitoring procedure can include the ordinary monitoring of BSs for potential handover candidates. It should be noted that monitoring of a known sequence from a BS can usually be performed even in cases where limited "hearability" prevents a SOHO with that BS.

At step **508**, the MS measures the TOA of downlink signals transmitted by several different BSs. Each TOA measurement can be measured in the time base of the MS or as a relative value to the source BS or to some other BS. The TOA measurements are temporarily stored in local memory along with information identifying the BS that corresponds to each TOA measurement. This data is then sent to the BSC for further processing. The measurements of step **508** can be made on the pilot channel data or the traffic channel data. Because the BSs generally "know" the offsets t_i, the time difference between the BSs (i.e., the time difference between the pilot frame transmissions of the BSs) is known even when traffic channel is used. At step **510**, the BSC adjusts the TOA measurements to account for the RTDs between the various BSs by adding the RTD estimates to the TOA measurements. At step **512**, a propagation delay time is calculated for each downlink signal using the adjusted TOA measurements, and the location of the MS is estimated at step **514** using the calculated propagation delay times. The positioning information can then, for instance, be transmitted to the MS, stored at the BSC, or sent to the Home Location Register (HLR). In an alternative embodiment, the calculations of steps **510**, **512**, and **514** can also be made in the MS, MSC, or some other location in the network.

The method **500** illustrated in FIG. **5** provides positioning estimates based upon measurements made at the MS of the TOA of a downlink signal. In another alternative embodiment, mobile positioning is determined using an uplink signal. The uplink solution is essentially the same as the downlink solution except that, instead of measuring the TOA of downlink signals at step **508**, TOA measurements are made at multiple BSs on an uplink signal that is transmitted by the MS. These uplink signal measurements are then provided to the BSC or the MSC, and adjusted TOA measurements and propagation delay times are calculated, as in steps **510** and **512** of the downlink solution method **500**.

As discussed above in connection with the standard RTD estimates, the uncertainty in the improved RTD estimates will also increase with time if the RTD estimate table for a particular BS is not updated. For positioning purposes, however, the required accuracy of the RTD estimates is much greater than in the context of cell-searches. Thus, the improved RTD estimates obtained during SOHO should be recent enough so that the clocks in the BSs have not drifted compared to each other. Otherwise, it will be difficult, if not impossible, to perform accurate mobile positioning determinations. Many of the same methods described above for addressing the uncertainty problem in the standard RTD estimate context can be used in a similar manner to address uncertainty in the improved RTD estimate context.

The method of obtaining an improved RTD estimate can also be used to further improve the cell-search process described in connection with FIGS. 1 and 2. In one preferred embodiment of the cell-search method 100 (see FIG. 1), the time difference reported by the MS in SOHO is directly used to calculate the RTD estimates; no information from the BS is required. Thus, referring again to FIG. 4, the standard RTD estimate is equal to an MS's receive time T_{mr1} of a downlink signal from a first base station BS1, subtracted from the MS's receive time T_{mr2} of a downlink signal from a second base station BS2. In FIG. 4, this value is illustrated by the time differential t_{diff}. The direct use of the time difference as reported by the MS offers a significant improvement over prior cell-search procedures and, in most cases, sufficiently reduces the complexity of the cell-search process to overcome the problems found in other potential positioning methods.

If greater accuracy is required, however, the timing uncertainty in the MSs that do SOHO searches can be further reduced by as much as fifty percent by using an improved RTD estimate that takes propagation delays into account. By using improved RTD estimates, the set of time delays that the MS has to search during the cell-search process is considerably decreased, especially when compared to prior art cell-search methods. The cell-search uncertainty interval will then depend upon the size of the cell and the amount of sectorization of the cell. For example, in a non-sectorized cell system having a cell radius of approximately 30 kilometers, the uncertainty is less than 300 microseconds, assuming that the position of the mobile can be estimated to within 3 cell radii. The normal search window for prior cell-search methods, in contrast, is about 10 milliseconds. Thus, the use of an improved estimated RTD provides a two-orders-of-magnitude improvement in the search complexity. The results are even better in cellular systems with smaller cells or with sectorized cells. It is also possible to decrease the uncertainty interval for the cell-search even further by estimating the round-trip delay between a target BS and the mobile that is performing the cell-search, especially in the case of sectorized cells. An estimated round-trip delay can easily be calculated from the data available when making RTD calculations or if the approximate location of the mobile is known.

Although several preferred embodiments of the method and apparatus of the present invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the embodiments disclosed. For example, measurements of the relative timing of BSs, made in accordance with the present invention, could also be used for the pseudo-synchronization of BSs. Thus, the invention is capable of numerous rearrangements, modifications and substitutions without departing from the spirit of the invention as set forth and defined by the following claims.

What is claimed is:

1. A method for estimating a relative timing of a plurality of base stations in an asynchronous mobile communications system, comprising the steps of:

receiving at a first mobile station a first downlink signal transmitted by a first one of said base stations and a second downlink signal transmitted by a second one of said base stations;

transmitting an uplink signal from the first mobile station to the first and second base stations; and

calculating an estimated relative time difference between the time base of said first base station and the time base of said second base station using receive times at the first and second base stations of said uplink signal, transmit times of the first downlink signal and the second downlink signal, and a time difference at the first mobile station between a receive time of the second downlink signal and the transmit time of said uplink signal, wherein the receive times of the uplink signal and the transmit times of the first and second downlink signals are in the time base of the base station transmitting or receiving the respective signal, said calculation accounting for propagation delays between said first mobile station and said first and second base stations.

2. The method of claim 1, further comprising the step of utilizing the estimated relative time difference to calculate at least one possible location of a second mobile station relative to at least one of said first and second base stations by determining a distance between the second mobile station and at least one of said first and second base stations.

3. The method of claim 2, further comprising the step of determining a location of the second mobile station by using a plurality of relative time differences between a plurality of different pairs of base stations to calculate a distance between the second mobile station and each of a plurality of said base stations.

4. The method of claim 1, further comprising the step of determining at least one possible location of a second mobile station using the receive times at said first base station and said second base station of an uplink signal transmitted by said second mobile station and using said estimated relative time difference.

5. The method of claim 1, further comprising the step of determining at least one possible location of a second mobile station using receive times at the second mobile station of downlink signals transmitted by each of said first and second stations and using said estimated relative time difference.

6. The method of claim 1, further comprising the step of transmitting the value of the estimated relative time difference to a second mobile station.

7. The method of claim 6, further comprising the steps of:

estimating a propagation delay of signals transmitted between the second mobile station and the first base station and of signals transmitted between the second mobile station and the second base station based on an approximate location of the second mobile station;

adjusting the estimated relative time difference value by factoring in the estimated propagation delays to determine a local estimated relative time difference value at the mobile station;

said second mobile station correlating said local estimated relative time difference value with a matched filter output signal; and

initiating a cell search based on a result of the correlating step.

8. The method of claim 6, further comprising the steps of:

said second mobile station correlating the estimated relative time difference value with a matched filter output signal; and

initiating a cell search based on a result of the correlating step.

9. The method of claim 1, wherein said estimated relative time differences are used to synchronize said first and second base stations.

10. The method of claim 1, further comprising the step of storing the computed difference in a relative time difference table.

11. The method of claim 1, wherein said first mobile station is in a state of handover.

12. A method for facilitating timing between mobile stations and base stations in an asynchronous mobile telecommunications network, comprising the steps of:

receiving relative timing difference data from each of a plurality of mobile stations, the relative timing difference data from each mobile station including a measured difference between the time bases of at least two base stations as measured by said mobile stations;

determining a relative timing difference estimate based on the received relative timing difference data, the relative timing difference estimate representing an estimate of a difference between the time bases of at least two base stations;

accounting for propagation delays between a measuring mobile station and the at least two base stations of which the time base difference is measured in said relative timing difference estimate;

storing the relative timing difference estimate in a relative timing difference table;

transmitting the relative timing difference estimate to a receiving mobile station.

13. The method of claim 12, wherein the relative timing difference estimate is determined by calculating a relative timing difference estimate value from a plurality of measured differences received from the plurality of mobile stations.

14. The method of claim 12 further comprising the steps of:

estimating an error range for the relative timing difference estimate; and

transmitting the error range to the receiving mobile station.

15. The method of claim 12, wherein the relative timing difference estimate transmitted to the receiving mobile station is used to estimate a position of the receiving mobile station.

16. The method of claim 12, wherein the relative timing difference estimate comprises the measured difference received from one of the plurality of mobile stations.

17. The method of claim 12, wherein the relative timing difference estimate transmitted to the receiving mobile station is used to assist in synchronizing the receiving mobile station with a cell.

18. The method of claim 12, wherein data from a neighbor cell list is transmitted along with the relative timing difference estimate.

19. An asynchronous mobile telecommunications system, comprising:

a plurality of base stations for transmitting data to and receiving data from a plurality of mobile stations, the plurality of base stations individually receiving relative timing difference data, the relative timing difference data from each mobile station comprising a measured difference between the time bases of two of the plurality of base stations as measured by the mobile station;

a register storing a relative timing difference table, each of a plurality of entries in said table comprising a relative timing difference estimate calculated from the relative timing difference data and said register storing error data for each relative timing difference estimate; and

wherein a first one of the plurality of base stations transmits a relative timing difference estimate to a receiving mobile station to facilitate timing of communications between the mobile station and a second one of the plurality of base stations.

20. The system of claim 19 wherein the register further stores a neighbor cell list.

21. A method for facilitating the timing of a plurality of base stations in an asynchronous mobile communications system, comprising the steps of:

at least one of said plurality of base stations sending at least one estimated relative time difference value to a mobile station, said at least one estimated relative time difference value comprising an estimated timing difference between said at least one of said plurality of base stations and a neighbor base station;

said mobile station receiving said at least one estimated relative time difference value;

said mobile station correlating said at least one estimated relative time difference value with a matched filter output signal; and

initiating a cell search based on a result of the correlating step.

22. The method of claim 21, wherein the correlating step comprises:

comparing said at least one estimated relative time difference value to said matched filter output signal; and

determining if said at least one estimated relative time difference value is likely to correspond to a matched filter output signal peak.

23. The method of claim 22, wherein the initiating step comprises selecting a scrambling code based on a result of the determining step.

24. The method of claim 21, further comprising the step of estimating a propagation delay between said mobile station and at least one of said base stations, said estimated propagation delay used to reduce uncertainty in said correlating step.

25. The method of claim 21, wherein said mobile station transmits said at least one estimated relative time difference value along with a neighboring-cell quality report to said at least one of said plurality of base stations.

26. The method of claim 25, wherein a base station controller associated with said at least one of said plurality of base stations stores said at least one estimated relative time difference in a database.

27. The method of claim 21, wherein the sending step comprises broadcasting or transmitting said at least one estimated relative time difference value in a neighbor list message.

28. The method of claim 27, wherein said neighbor list message includes at least-one scrambling code associated with said neighbor base station.

29. The method of claim 21, wherein the sending step further comprises sending an uncertainty value associated with said at least one estimated relative time difference value.

30. The method of claim 21, wherein the mobile communications system comprises an asynchronous DS-CDMA system.

31. A method for facilitating the timing of a plurality of base stations in an asynchronous mobile communications system, comprising the steps of:

at least one of said plurality of base stations sending at least one estimated relative time difference value to a mobile station, said at least one estimated relative time difference value comprising an estimated timing difference between said at least one of said plurality of base stations and a neighbor base station;

said mobile station receiving said at least one estimated relative time difference value; and

determining an approximate position of the mobile station using said at least one estimated relative time difference value.

32. The method of claim 31 wherein said step of determining an approximate position comprises the steps of:

calculating a time difference at the mobile station between receive times of a first downlink signal transmitted by said at least one of said plurality of base stations and a second downlink signal transmitted by said neighbor base station; and

comparing said at least one relative time difference value with said calculated time difference to determine at least one possible location of said mobile station relative to said at least one of said plurality of base stations and said neighbor base station.

33. A system for synchronizing a plurality of base stations in a mobile communications system, comprising:

a first base station of said plurality of base stations, said first base station operable to broadcast or transmit at least one estimated relative time difference value, said at least one estimated relative time difference value comprising an estimated timing difference between said first base station and a neighbor base station;

a mobile station for receiving said at least one estimated relative time difference value; and

a processor for determining an approximate position of the mobile station using said at least one estimated relative time difference value.

34. The system of claim 33, wherein said processor compares said at least one estimated relative time difference value with a time difference measured by said mobile station to determine at least one possible location of said mobile station relative to said first base station and said neighbor base station.

35. A system for synchronizing a plurality of base stations in a mobile communications system, comprising:

a first base station of said plurality of base stations, said first base station operable to broadcast or transmit at least one estimated relative time difference value and at least one uncertainty value associated with said at least one estimated relative time difference value, said at least one estimated relative time difference value comprising an estimated timing difference between said first base station and a neighbor base station; and

a mobile station for receiving said at least one estimated relative time difference value and said at least one uncertainty value.

36. The system of claim 35, wherein said mobile station is operable to correlate said at least one estimated relative time difference value with a matched filter output signal and initiate a cell search.

37. The system of claim 36, wherein said mobile station is operable to compare said at least one estimated relative time difference value to said matched filter output signal, and determine if said at least one estimated relative time difference value is likely to correspond to a matched filter output signal peak.

38. The system of claim 36, wherein said mobile station is operable to select a scrambling code based on a result of correlating said at least one estimated relative time difference value with a matched filter output signal peak.

39. The system of claim 35, wherein said mobile station transmits said at least one estimated relative time difference value along with a neighboring-cell quality report to said first base station.

40. The system of claim 35, wherein a base station controller associated with said first base station stores said at least one estimated relative time difference in a database.

41. The system of claim 35, wherein said first base station is operable to broadcast or transmit said at least one estimated relative time difference value in a neighbor list message.

42. The system of claim 35, wherein the mobile communications system comprises an asynchronous DS-CDMA system.

* * * * *

US006389019B1

(12) **United States Patent**
Fan et al.

(10) **Patent No.: US 6,389,019 B1**
(45) **Date of Patent: May 14, 2002**

(54) **TIME-BASED SCHEDULER ARCHITECTURE AND METHOD FOR ATM NETWORKS**

(75) Inventors: **Ruixue Fan**, Plainsboro; **Brian L. Mark**, Princeton; **Gopalakrishan Ramamurthy**, Cranbury; **Alexander T. Ishii**, Princeton, all of NJ (US)

(73) Assignee: **NEC USA, Inc.**, Princeton, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/040,311**

(22) Filed: **Mar. 18, 1998**

(51) **Int. Cl.**[7] **H04L 12/56**
(52) **U.S. Cl.** **370/395**; 370/233
(58) **Field of Search** 370/230, 352, 370/257, 419, 389, 440, 432, 397, 395, 232, 229, 231–237, 241, 251–254, 387, 337, 447, 445, 442, 400, 412, 413, 428, 418, 351; 395/200.3

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,706,288 A	* 1/1998	Radhakrishnan et al.	...	370/418
5,719,853 A	* 2/1998	Ikeda	...	370/229
5,751,709 A	5/1998	Rathnavelu		
5,774,653 A	* 6/1998	Katevenis	...	395/200.3
5,838,677 A	* 11/1998	Kozaki et al.	...	370/389
5,850,395 A	* 12/1998	Hauser et al.	...	370/398
5,864,540 A	* 1/1999	Bonomi et al.	...	370/235
5,889,779 A	* 3/1999	Lincoln	...	370/398
5,892,762 A	4/1999	Okuda et al.		
5,909,443 A	* 6/1999	Fichou et al.	...	370/412
6,005,866 A	* 12/1999	Linoln	...	370/398
6,011,775 A	* 1/2000	Bonomi et al.	...	370/230
6,028,843 A	* 2/2000	Delp et al.	...	370/235
6,226,265 B1	* 5/2001	Nakamichi et al.	...	370/235

FOREIGN PATENT DOCUMENTS

JP	9-36912	2/1997
JP	9-200231	7/1997
WO	WO 97/14240	4/1997

OTHER PUBLICATIONS

Yoshihiro Ohba, "QLWFQ: A Queue Length Based Weighted Fair Queueing Algorithm in ATM Networks," IEEE, Apr. 7, 1997, pp. 566–575.
Jennifer Rexford, et al., "A Scalable Architecture for Fair Leaky–Bucket Shaping," IEEE Infocom '97, Proceedings vol. 3, pp. 1054–1062.

* cited by examiner

Primary Examiner—Dang Ton
(74) *Attorney, Agent, or Firm*—Sughrue Mion, PLLC

(57) **ABSTRACT**

A flexible and scalable architecture and method that implements dynamic rate control scheduling in an ATM switch. The scheduler shapes a large number of streams according to rate values computed dynamically based on switch congestion information. To handle a large range of bit rates, a plurality of timewheels are employed with different time granularities. The streams are assigned dynamically to the timewheels based on computed rate values. The shaper architecture and method support priority levels for arbitrating among streams which are simultaneously eligible to transmit.

30 Claims, 17 Drawing Sheets

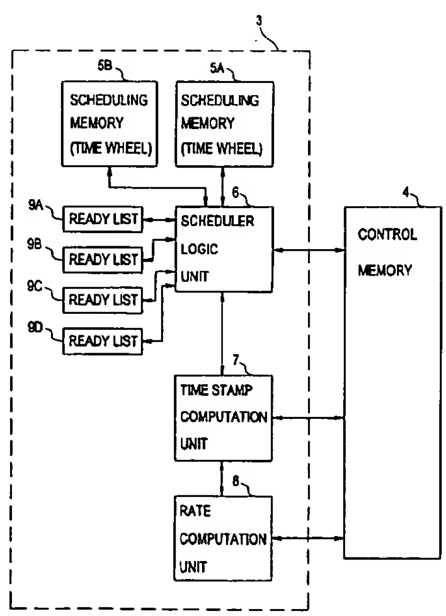


FIG. 1

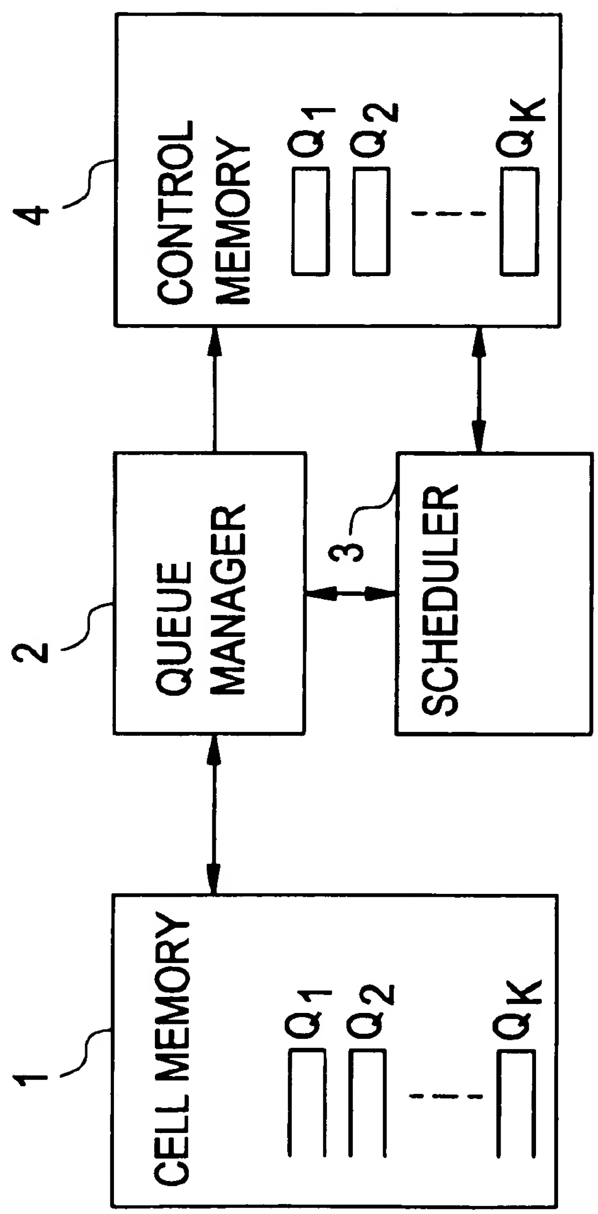

4
CONTROL MEMORY
Q1
Q2
QK
2
QUEUE MANAGER
3
SCHEDULER
1
CELL MEMORY
Q1
Q2
QK

FIG. 2

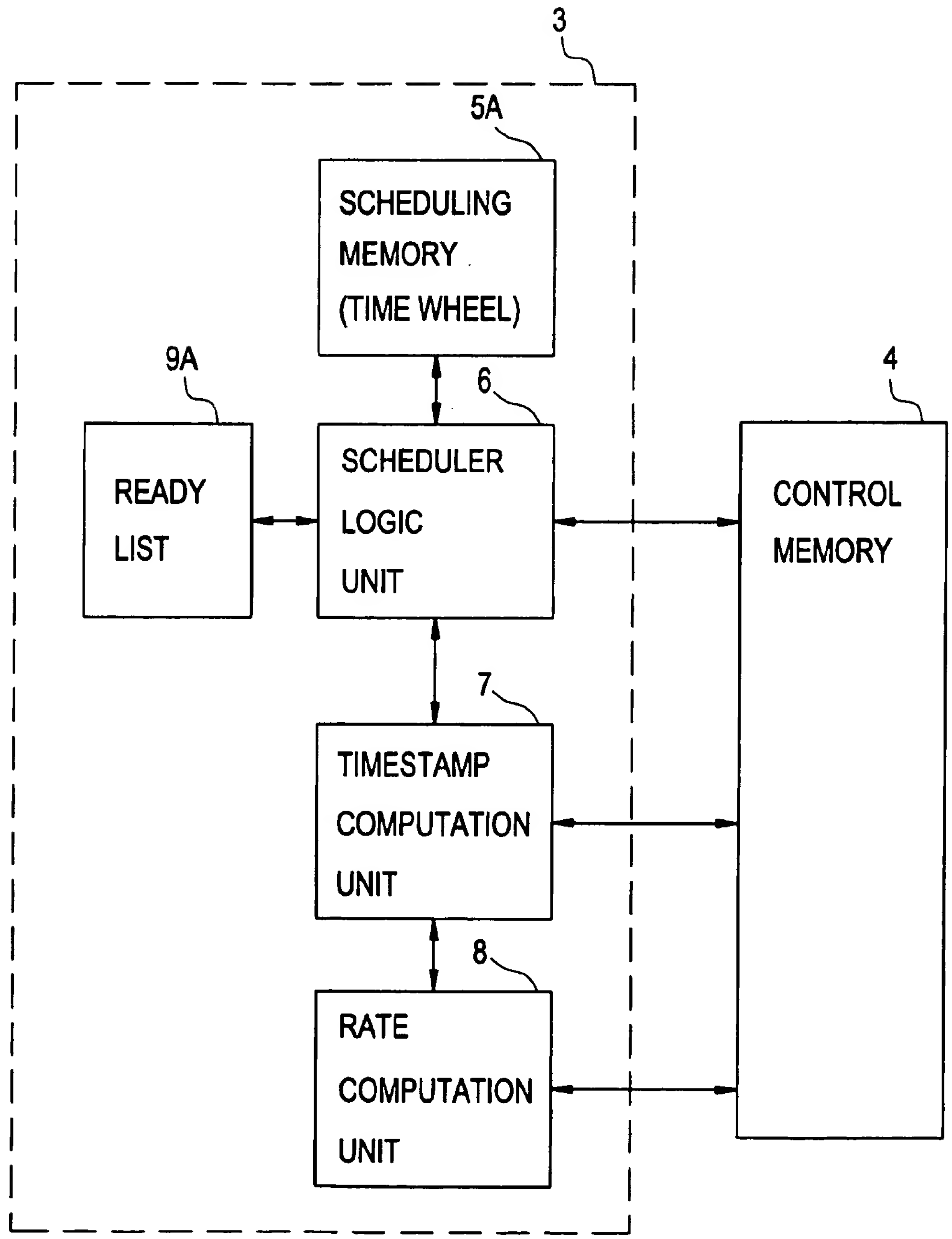
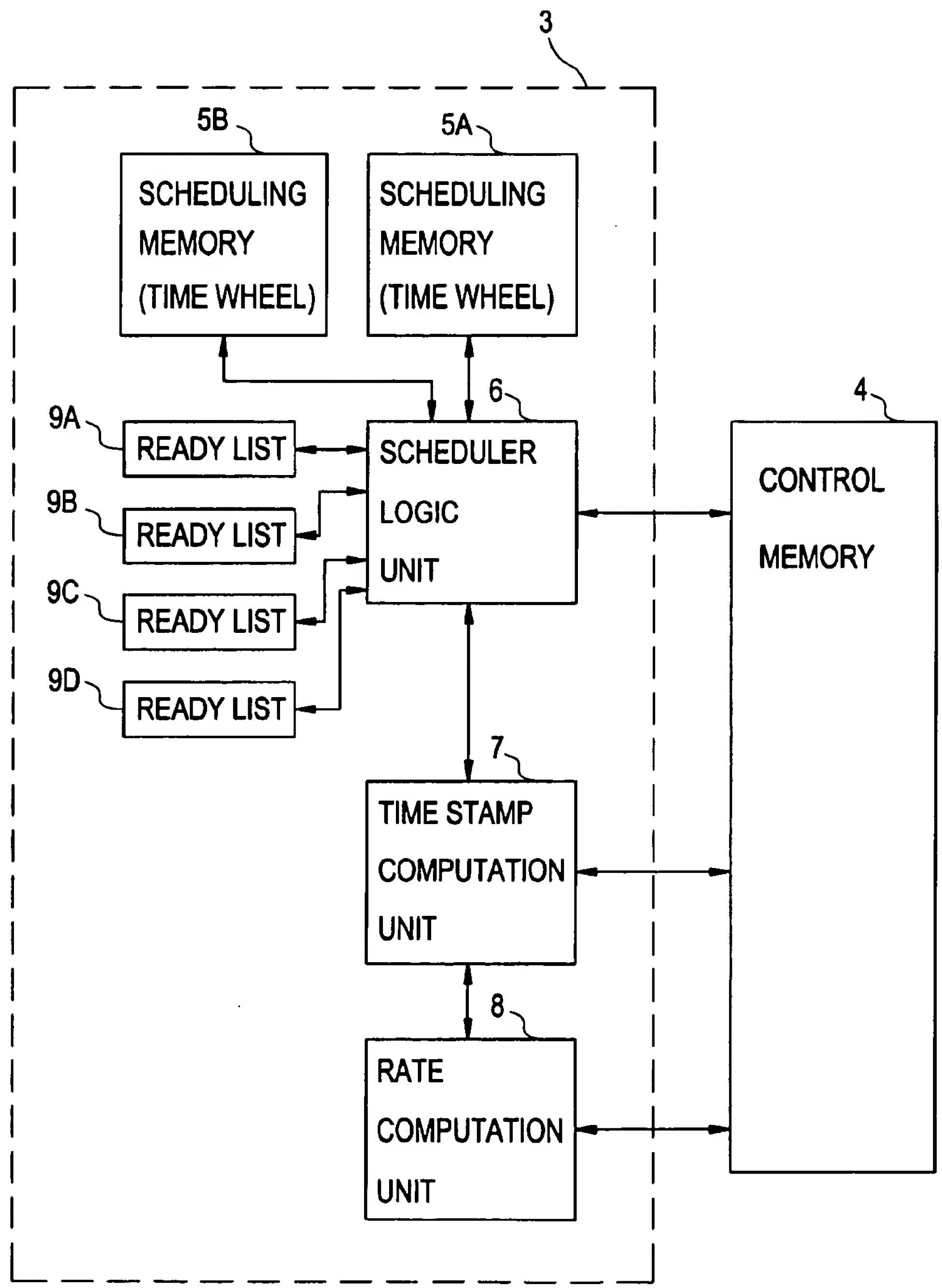

3
5A
SCHEDULING MEMORY (TIME WHEEL)
9A
6
4
READY LIST
SCHEDULER LOGIC UNIT
CONTROL MEMORY
7
TIMESTAMP COMPUTATION UNIT
8
RATE COMPUTATION UNIT

FIG. 3



3
5B
SCHEDULING MEMORY (TIME WHEEL)
5A
SCHEDULING MEMORY (TIME WHEEL)
6
4
9A
READY LIST
9B
READY LIST
9C
READY LIST
9D
READY LIST
SCHEDULER LOGIC UNIT
CONTROL MEMORY
7
TIME STAMP COMPUTATION UNIT
8
RATE COMPUTATION UNIT

FIG. 4

CELL ARRIVAL TIMESTAMP COMPUTATION

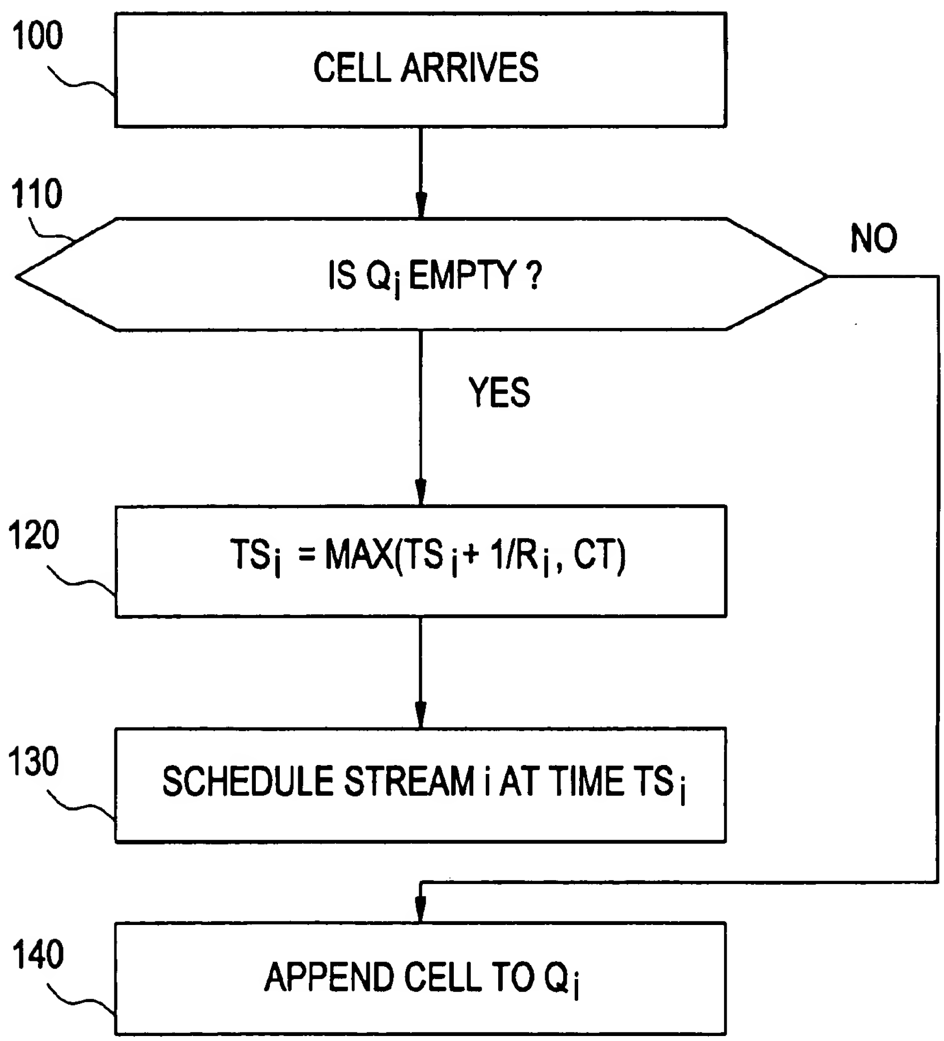


FIG. 5

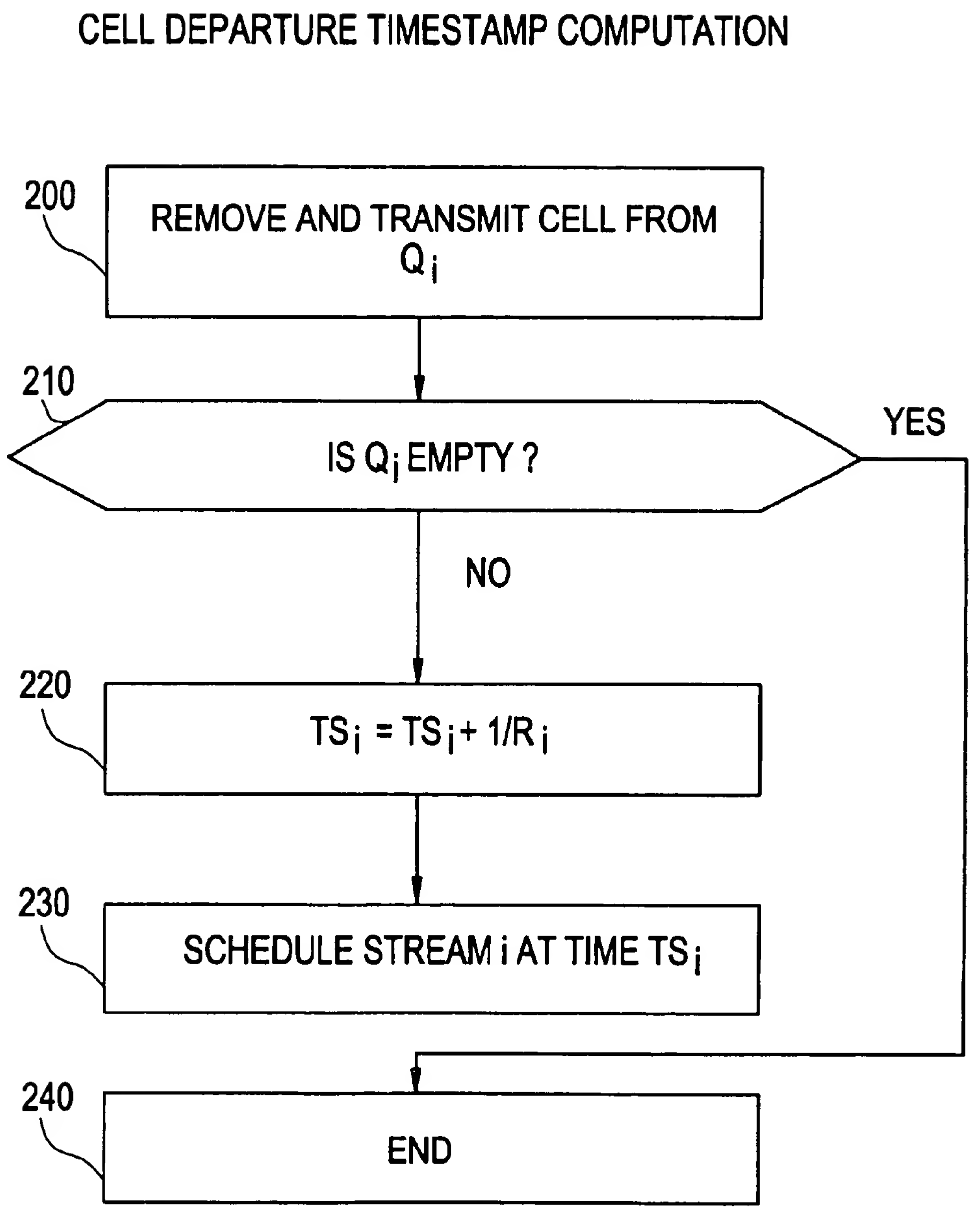

CELL DEPARTURE TIMESTAMP COMPUTATION
200
REMOVE AND TRANSMIT CELL FROM Q_i
210
IS Q_i EMPTY ?
YES
NO
220
$TS_i = TS_i + 1/R_i$
230
SCHEDULE STREAM i AT TIME TS_i
240
END

FIG. 6

PROCEDURE FOR SETTING IDLE BIT

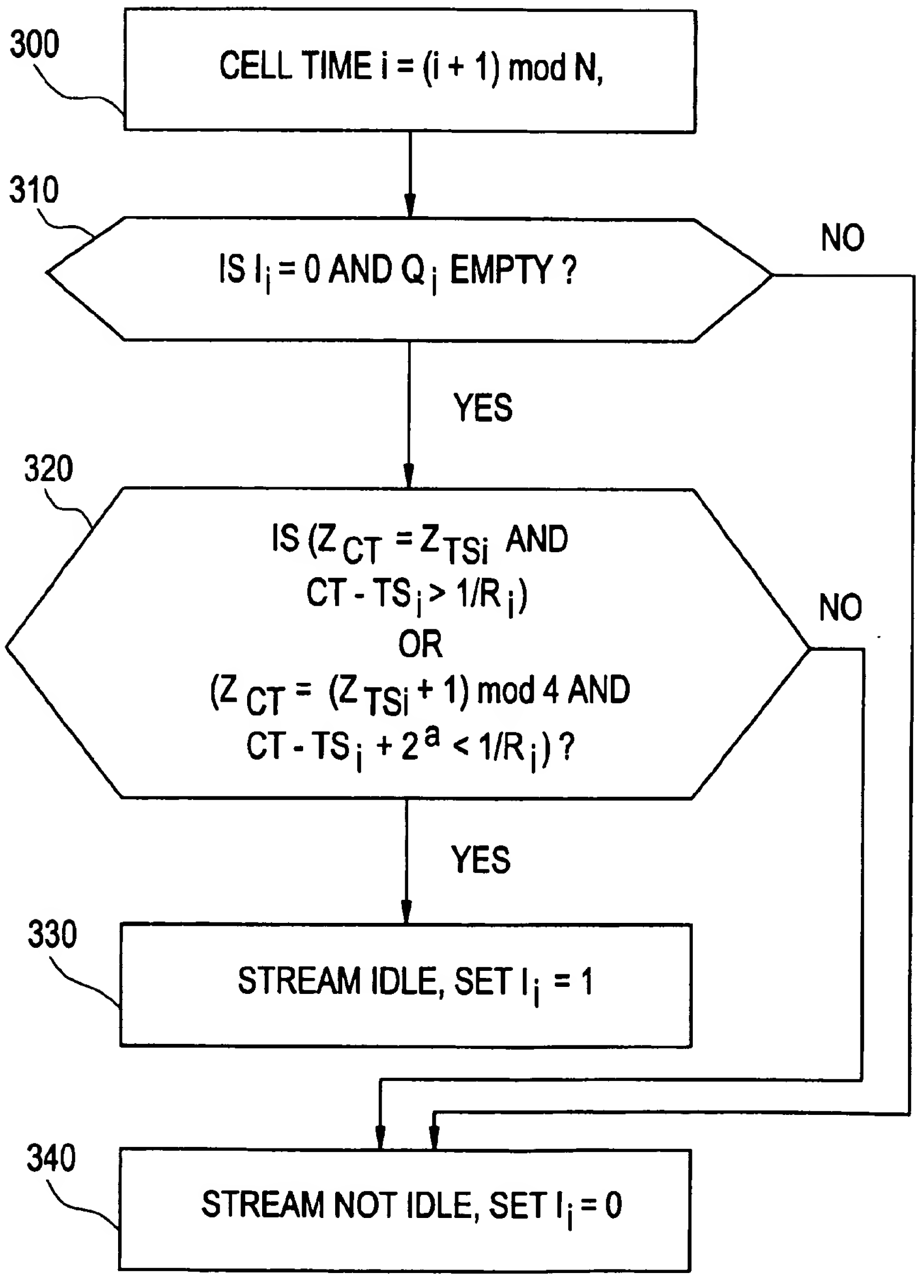


FIG. 7

PROCEDURE INCORPORATING WRAP AROUND

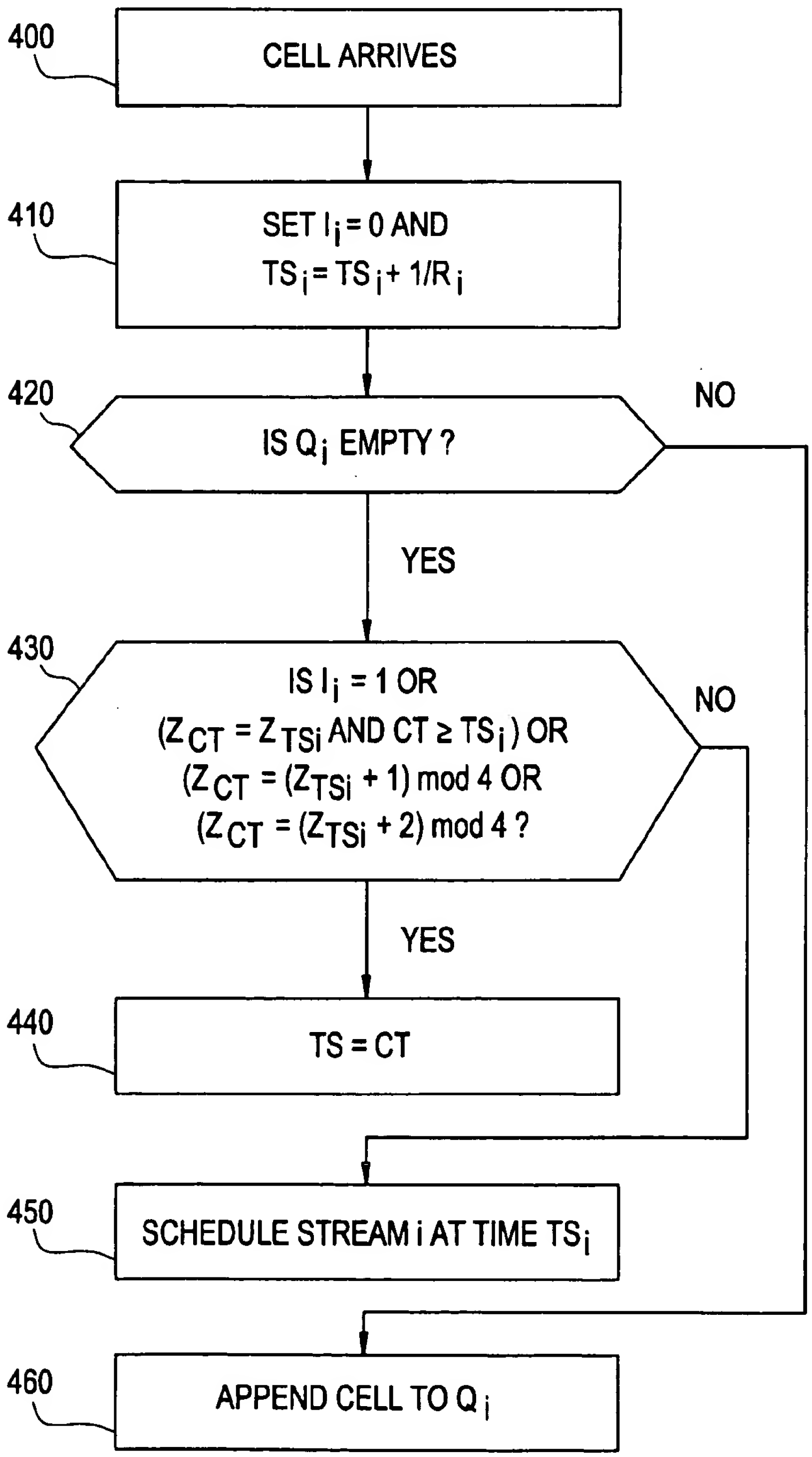


FIG. 8

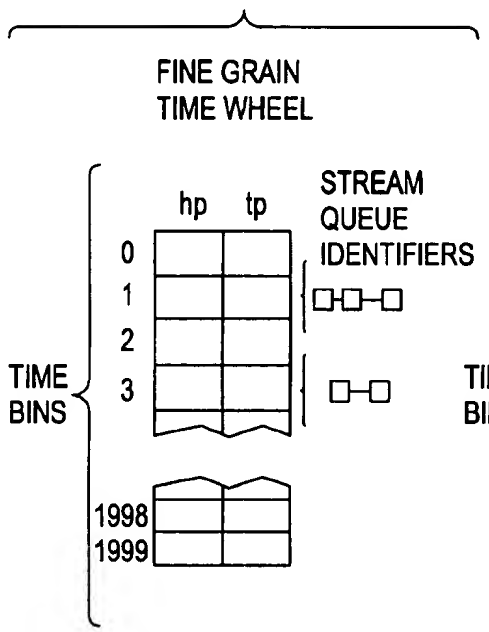


FIG. 9

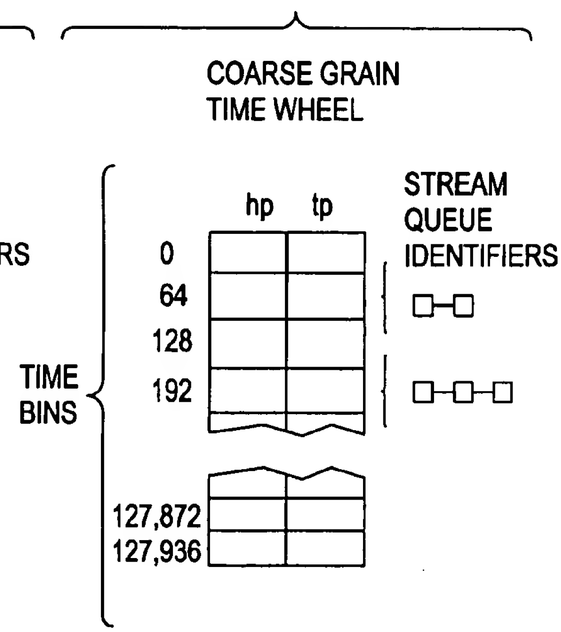


FIG. 10

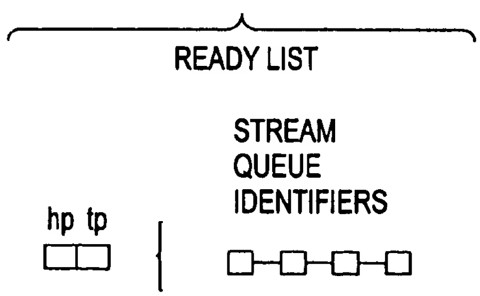


FIG. 11

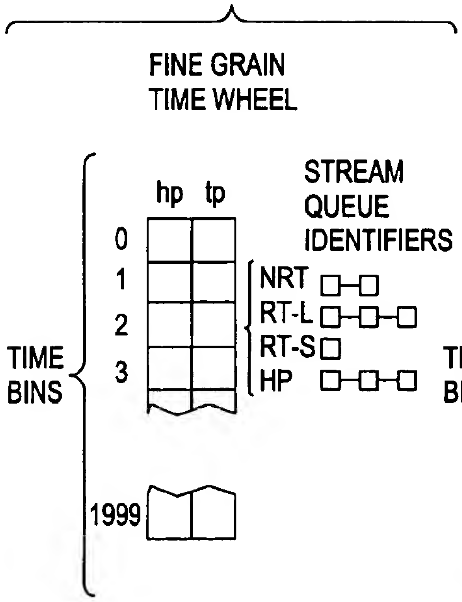


FIG. 12

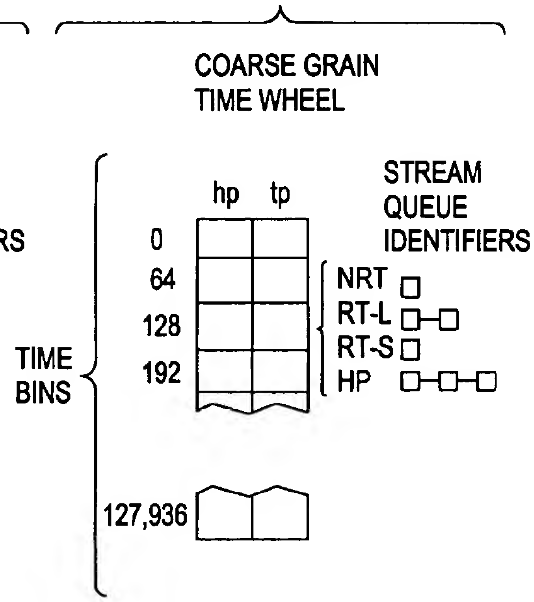


FIG. 13

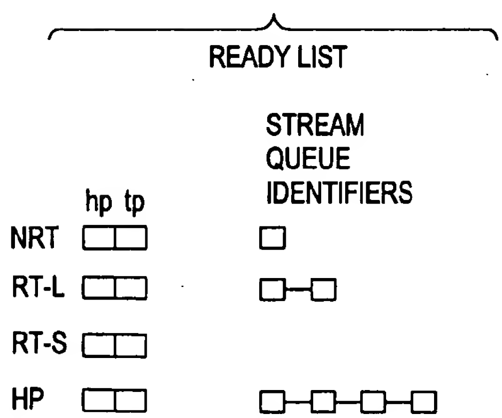


FIG. 14

PROCEDURE FOR ATTACHING
A STREAM QUEUE IDENTIFIER
TO A READY LIST

```
if = CT = 0 mod 64, then
      X = C; t = CT/64
else
      X = F; t = CT mod 2K
end if

rc =  rc + M_X[ t ];
for i = 0 to 3 do
      if B_X [ i ] [ t ] = 1, then
            tp = T_X[ i ] [ t ].tp; hp = T_X[ i ] [ t ].hp
            if r[ i ].hp = 0, then
                  r[ i ].hp = hp; r[ i ].tp = tp
            else
                  v[ r [ i ].tp ] = hp; r[ i ].tp = tp
            end if
            B_X [ i ] [ t ] = 0
      end if
end for
```

FIG. 15

PROCEDURE FOR INSERTING A STREAM QUEUE IDENTIFIER

```
if (Z_TS = Z_CT and TS < CT) or (Z_TS ≠ Z_CT + 1) then
    if CT = 0 mod 64, then
        X = C; t = CT/64;
    else
        X = F; t = CT mod 2K;
    end if
else
    if Z_TS = Z_CT, then
        if TS - CT < 2K, then
            t = TS mod 2K; X = F
        else
            t = TS/64; X = C
        end if
    else
        if 128K - (CT - TS) and (TS mod 64 ≠ 0), then
            X = F; t = TS mod 2K;
        else
            X = C; t = TS/64
        end if

    end if

if B_X[ i ] [ t ] = 0, then
    B_X[ i ] [ t ] = 1;
    M_X[ t ] = 1;
    T_X[ i ] [ t ].hp = stream queue identifier;
    T_X[ i ] [ t ].tp = stream queue identifier;

else
    tp = T_X[ i ] [ t ].tp
    M_X[ t] = M_X[ t ] + 1;
    T_X[ i ] [ t ].tp = stream queue identifier
    V[ tp ] = stream queue identifier

end if
```

FIG. 16

PROCEDURE FOR EXTRACTING A STREAM QUEUE IDENTIFIER FROM A READY LIST

```
for i = 0 to 3, do
    while (sufficient time in cell slot) and
    (r[ i ].hp ≠ 0), then
        if r[ i ].hp = r[ i ].tp, then
            q = r[ i ].hp
            r[ i ].hp = 0
        else
            q = r[ i ].hp
            r[ i ].hp = V[ e ]
        end if
        rc = rc - 1;
        Pass q on to another process to serve the
        queue
    end while
end for
```

FIG. 17 CELL ARRIVAL TIMESTAMP COMPUTATION COMBINING SCHEDULING AND UPC SHAPING

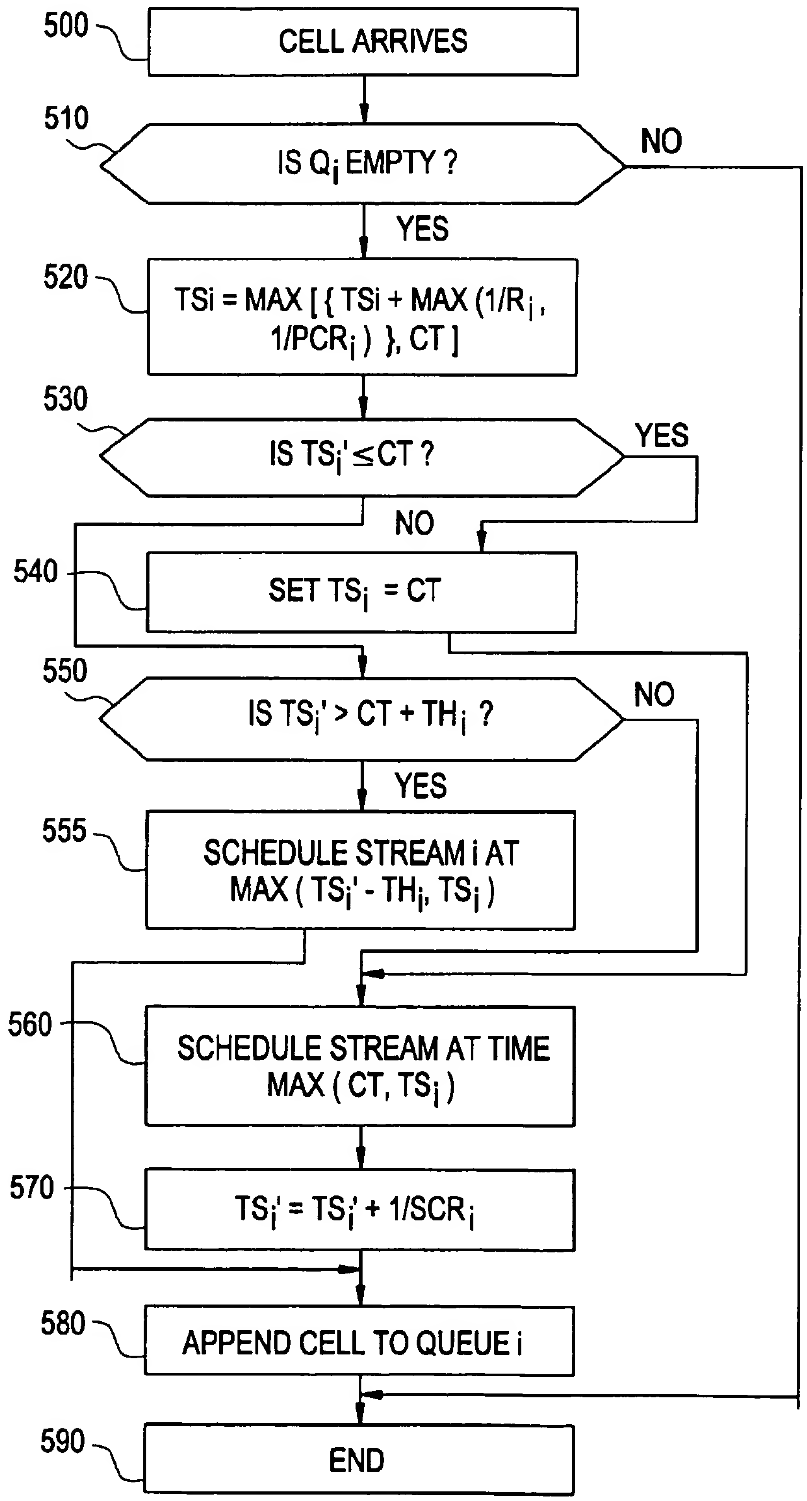


FIG. 18

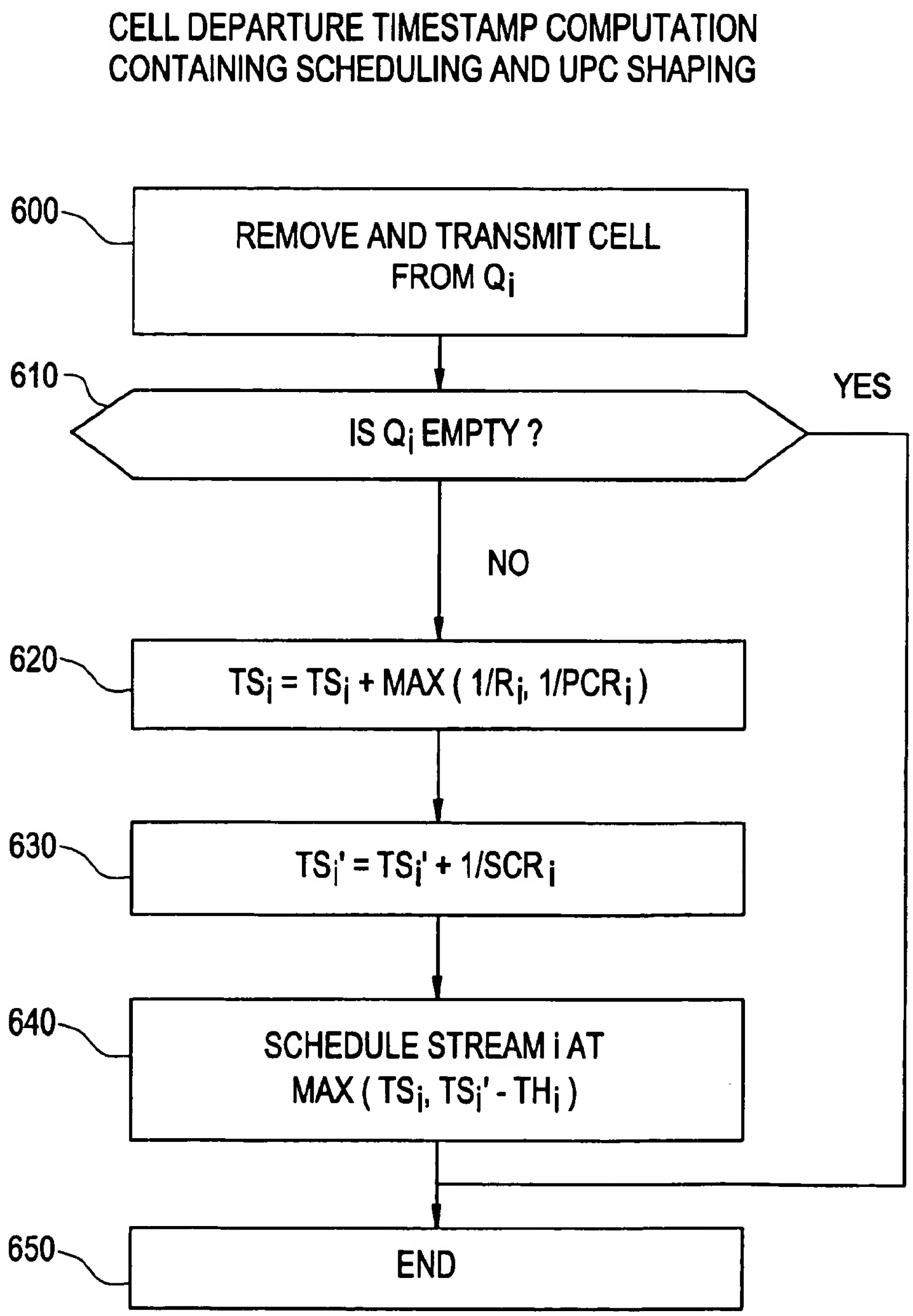

CELL DEPARTURE TIMESTAMP COMPUTATION
CONTAINING SCHEDULING AND UPC SHAPING
600
REMOVE AND TRANSMIT CELL FROM Q_i
610
IS Q_i EMPTY ?
YES
NO
620
$TS_i = TS_i + MAX(1/R_i, 1/PCR_i)$
630
$TS_i' = TS_i' + 1/SCR_i$
640
SCHEDULE STREAM i AT $MAX(TS_i, TS_i' - TH_i)$
650
END

FIG. 19

Fine Grain Time Wheel

Number of priority levels L = 4
Granularity of coarse wheel G = 64

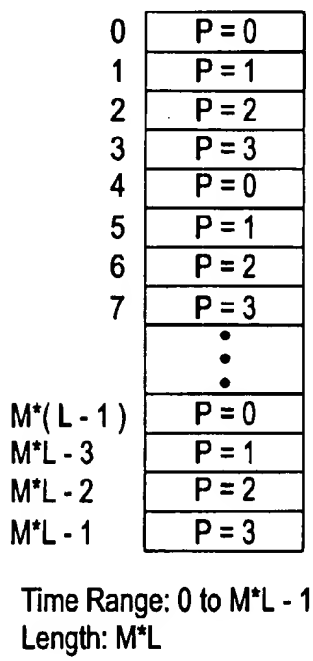


FIG. 20

Coarse Grain Time Wheel

Number of priority levels L = 4
Granularity of coarse wheel G = 64

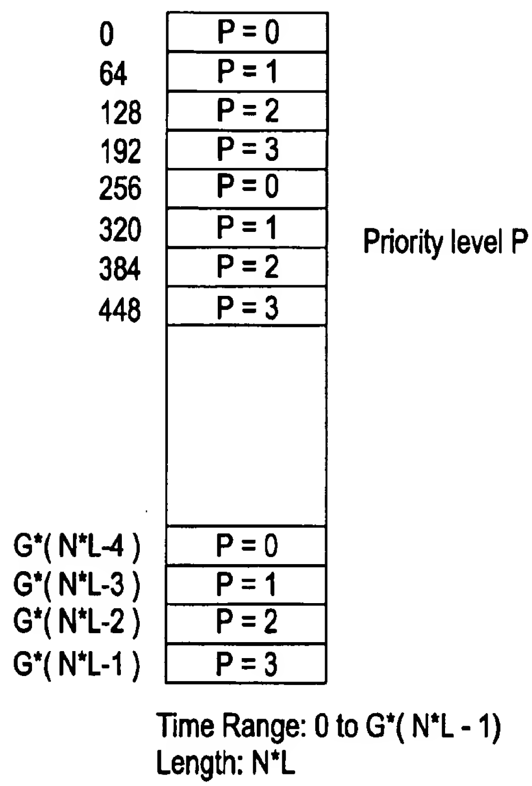


FIG. 21

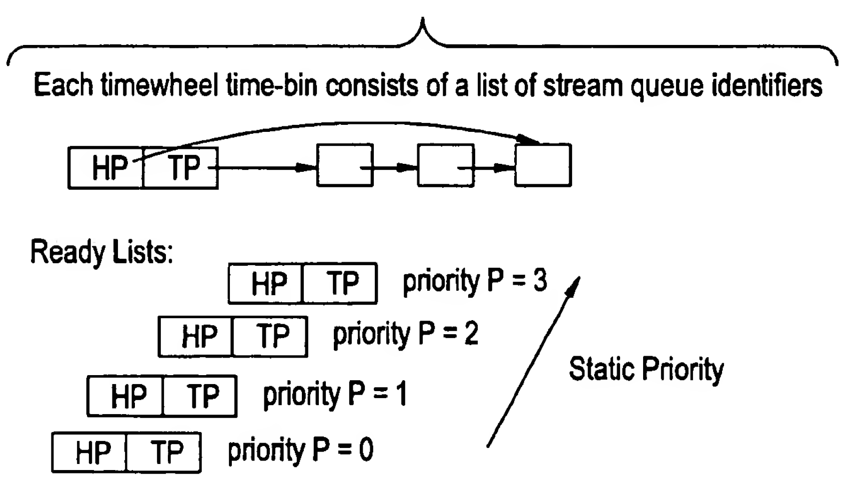


FIG. 22

Timewheel scheduling operations

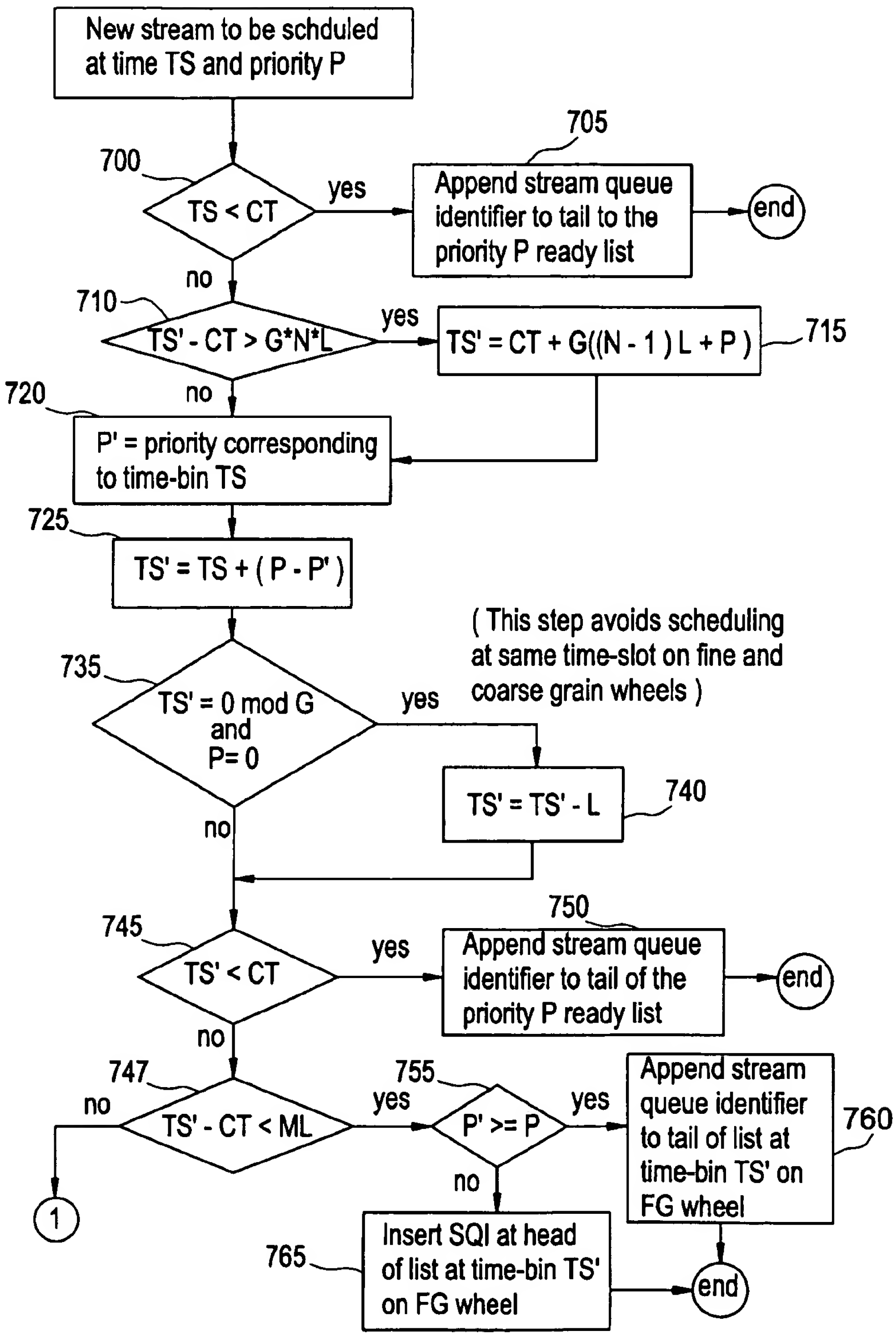
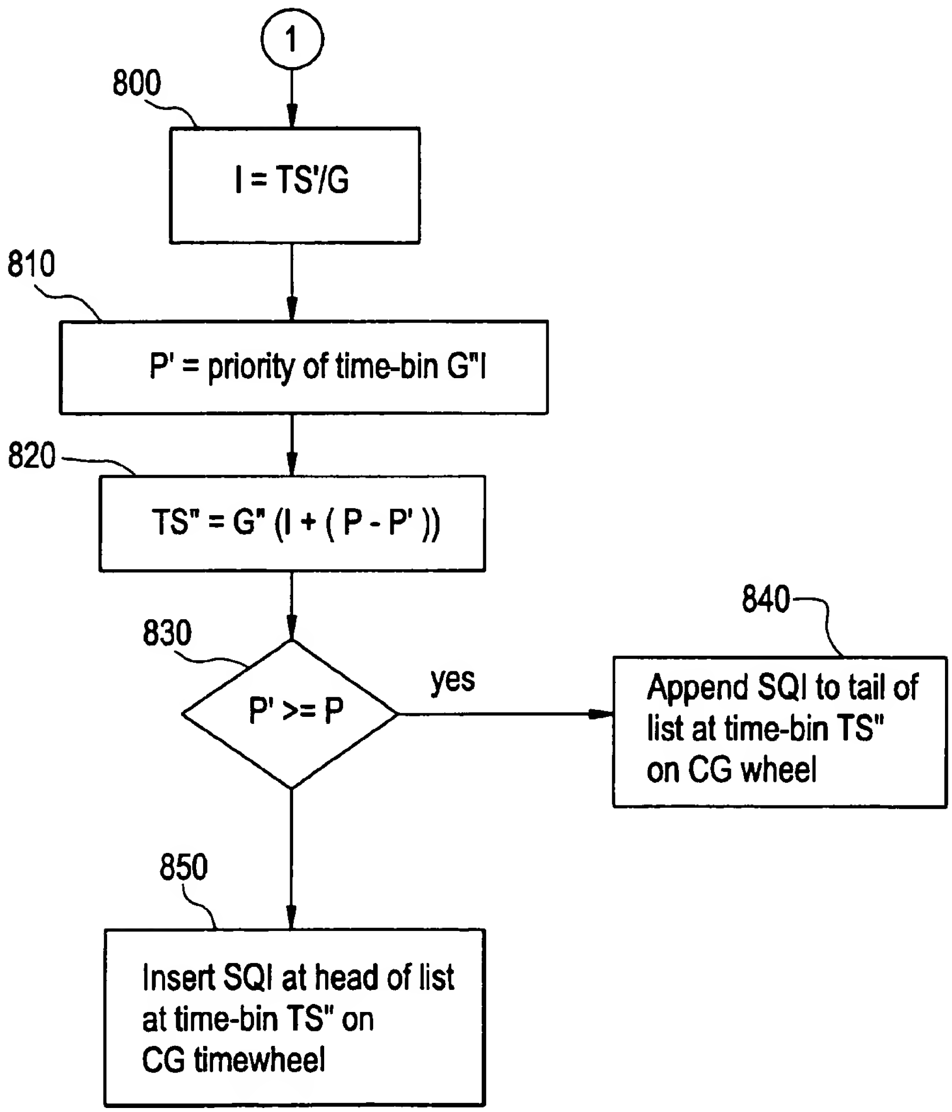


FIG. 23

Timewheel scheduling operations (continued)



TIME-BASED SCHEDULER ARCHITECTURE AND METHOD FOR ATM NETWORKS

This application relates to U.S. application Ser. No. 08/924,820 filed on Sep. 5, 1997 entitled, "Dynamic Rate Control Scheduler for ATM Networks," and Ser. No. 08/923,978, now U.S. Pat. No. 6,324,165, filed on Sep. 5, 1997 entitled, "Large Capacity, Multiclass Core ATM Switch Architecture," both of which are assigned to the Assignee of the present invention and which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to schedulers for asynchronous transfer mode (ATM) networks and, more specifically, to an architecture and method for scheduling stream queues serving cells with different quality-of-service (QoS) requirements while shaping the transmission rate to avoid congestion at bottlenecks within an ATM switch.

2. Description of Related Art

The function of a scheduler is to determine the order in which cells queued at a port are to be sent out. The simplest scheduling method is a first-in, first-out (FIFO) method. Cells are buffered in a common queue and sent out in the order in which they are received. The problem with FIFO queuing is that there is no isolation between connections or even between traffic classes. A "badly behaving" connection (i.e., it sends cells at a much higher rate than its declared rate) may adversely affect quality of service (QoS) of other "well behaved" connections.

A solution to this problem is to queue cells in separate buffers according to class. One further step is to queue cells on a per connection basis. The function of the scheduler is to decide the order in which cells in the multiple queues should be served. In round-robin (RR) scheduling, the queues are visited in cyclic order and a single cell is served when a visited queue is not empty. However, if all queues are backlogged, the bandwidth is divided equally among the queues. This may not be desirable, however, because queues may be allocated different portions of the common link bandwidth.

In weighted round-robin (WRR) scheduling, which was described in a paper by Manolis Katevenis, et al., entitled, "Weighted Round-Robin Cell Multiplexing in a General Purpose ATM Switch Chip," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, pp. 1265–1279, October 1991, each queue (connection or class queue) is assigned a weight. WRR aims to serve the backlogged queues in proportion to the assigned weights. WRR is implemented using counters, one for each queue. The counters are initialized with the assigned weights. A queue is eligible to be served if it is not empty and has a positive counter value. Whenever a queue is served, its counter is decreased by one (to a minimum of zero). Counters are reset with the initial weights when all other queues are either empty or have zero counter value. One problem with this counter-based approach is that the rate granularity depends on the choice of frame size (i.e., the sum of weights).

Another method, weighted fair queuing (WFQ), also known as packet-by-packet generalized sharing (PGPS), was described in a paper by Alan Demers, et al., entitled, "Analysis and Simulation of a Fair Queuing Algorithm," Proc. SIGCOMM'89, pp. 1–12, Austin, Tex., September 1989, and a paper by S. Jamaloddin Golestani, entitled, "A Self-clocked Fair Queuing Scheme for Broadband Applications," IEEE, 0743-166X/94, 1994, pp. 5c.1.1–5c.1.11. This method is a scheduling algorithm based on approximating generalized processor sharing (GPS). In the GPS model, the traffic is assumed to be a fluid, such that the server can drain fluid from all queues simultaneously at rates proportional to their assigned weights. A timestamp is computed when each cell arrives. The value of the timestamp represents the finishing time of the cell in the fluid model. The WFQ method schedules by selecting the cell with the smallest timestamp value.

All the methods described above are work-conserving with respect to the local link bottleneck, in the sense that if there are cells in the buffer(s), one cell will be served during a cell time. In contrast, another cell scheduling scheme, dynamic rate control (DRC), which was developed in co-pending application Ser. No. 08/924,820, is in general, non-work conserving. A cell may be held back if it could cause congestion downstream. DRC scheduling uses timestamps, as in WFQ, but the timestamps represent absolute time values. Thus, DRC may hold back a cell, if necessary, to alleviate congestion at a later switch bottleneck. This feature cannot be achieved with WFQ or WRR. One feature of DRC is that it does not require sorting of the timestamps, since the timestamps are compared to an absolute time clock. Also, traffic shaping can easily be incorporated into the DRC scheduler.

SUMMARY OF THE INVENTION

The present invention is a flexible and scalable architecture and method that implements DRC scheduling. Details on the algorithms and principles underlying DRC scheduling, are described in co-pending application Ser. No. 08/924,820. A key component of the DRC scheduler is a traffic shaper that shapes multiple traffic streams based on dynamically computed rates. The rates are computed based on congestion information observed at switch bottlenecks. Alternatively, the rates can be computed based only on the congestion observed at the local bottleneck. The modular design of the scheduler allows it to be used in a variety of switch configurations. In particular, the DRC scheduler architecture and method of the present invention can be applied to the input-output buffered switch architecture discussed in co-pending application Ser. No. 08/923,978 now U.S. Pat. No. 6,324,165.

The traffic shaper can shape a large number of streams with a wide range of associated rate values. With current technology, the architecture is able to support per VC queuing with up to 64 K virtual channels (VCs) with bit rates ranging from 4 Kbps to 622 Mbps. Scalability with respect to the number of streams that can be supported is achieved by scheduling streams to be served using a timewheel data structure, also known as a calendar queue. Calendar queues are well known. See for example, an article by R. Brown entitled, "Calendar Queues: A Fast 0(1) Priority Queue Implementation for the Simulation Event Set Problem," Communications of the ACM, Vol. Oct. 31, 1988, which is incorporated herein by reference.

To handle a large range of bit rates, a plurality of timewheels are employed with different time granularities. The timewheel concept and the partitioning of rates into ranges are also well known. See for example, an article by J. Rexford, et al. entitled, "Scalable Architecture for Traffic Shaping in High Speed Networks, IEEE INFOCOM '97, (Kobe), April 1997, which is incorporated herein by reference. The shaper architecture of the present invention differs from the one described in the Rexford article in that it

supports priority levels for arbitrating among streams which are simultaneously eligible to transmit. The highest priority level is assigned dynamically to provide short time-scale minimum rate guarantees in DRC scheduling. The remaining priority levels provide coarse QoS differentiation for defining traffic classes. Also in this architecture, the assignment of streams to timewheels is dynamic, depending on the current rate value computed for the stream.

A primary object of the invention is to provide an architecture and method capable of scheduling stream queues serving cells with different QoS requirements while shaping the transmission rate to avoid congestion at bottlenecks in an ATM switch.

Another object of the invention is to provide a scheduler architecture that can be used to implement available bit rate (ABR) service virtual source (VS)/virtual destination (VD) protocols as outlined in "Traffic Management Specification, Version 4.0," The ATM Forum, March 1996).

Another object of the invention is to provide a scheduler architecture that performs both scheduling and dual leaky bucket usage parameter control (UPC) shaping as also outlined in "Traffic Management Specification, Version 4.0." UPC shaping is used to force a traffic stream to conform to UPC parameters in order to avoid cell tagging or discarding at the interface to another subnetwork through which the stream passes.

Herein, the principles of the present invention will be schematically described in consideration of the above to facilitate the present invention.

Briefly, the gist of the present invention resides in the fact that a dynamic rate is calculated in consideration of congestion information on a downstream side and a timestamp is calculated on the basis of the dynamic rate to schedule/reschedule a queue. More specifically, when the timestamp is denoted by TS, a timestamp for scheduling is given by max(TS+1/R, CT) while a timestamp for rescheduling is given by TS=TS+1/R where CT is a current time and R is the dynamic rate. Herein, it is to be noted that the dynamic rate R is calculated by R=M+wE where M and w are representative of a minimum guaranteed rate and a weight factor, respectively, and E is representative of an excess rate calculated on the basis of congestion information.

As readily understood from the above, the dynamic rate R depends on the excess rate E and is successively updated. In addition, the timestamps for scheduling/rescheduling are determined by the use of the most recently computed value of the dynamic rate R. This shows that the timestamps for scheduling/rescheduling are calculated in consideration of the congestion information.

The above-mentioned formulas related to scheduling/rescheduling can be modified to make each stream from the queue conform to UPC parameters, such as PCR (Peak Cell Rate), SCR (Sustainable Cell Rate), and MBS (Maximum Burst Size). For example, let the timestamps TS for scheduling/rescheduling be calculated so that they conform to the PCR. In this event, the timestamps TS for scheduling/rescheduling are given by TS=max (TS+max (1/R, 1/PCR), CT) and TS=TS+max (1/R, 1/PCR), respectively. From this fact, it is readily understood that each cell is transmitted with a time interval of at least 1/PCR which is left between two adjacent ones of the cells and which is specified by a shaping timestamp determined on the basis of the timestamp TS for scheduling/rescheduling. This shows that the cell stream will conform to policing of the peak cell rate (PCR) at the next downstream switch. Hence, the downstream policing mechanism will neither tag nor discard the cells in the shaped cell stream. For example, a CLP (Cell Loss Priority) tag may not be put into a logic state of "1" in the present invention.

This is true of the SCR also. On policing the SCR, a timestamp for transmitting a next following cell is practically calculated with reference to the SCR value and a predetermined burst threshold (TH) that is determined by the value of MBS (Maximum Burst Size).

At any rate, the above-mentioned method according to the present invention realizes shaping operation. In other words, a scheduler according to the present invention can execute not only scheduling/rescheduling but also shaping.

Alternatively, the method according to the present invention may be used in combination with an ABR virtual source (VS) which executes traffic shaping to force a stream to conform to the requirements of ABR. In this event, a queue is shaped according to the rate determined by an ABR mechanism (along with the dynamic scheduling rate).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a diagram of the main components of an ATM buffer module serving a switch input or output port.

FIG. **2** is a diagram of one embodiment of the scheduler architecture of the present invention.

FIG. **3** is a diagram of another embodiment of the scheduler architecture of the present invention.

FIG. **4** is a flow chart showing the procedure for computing a timestamp value when a cell arrives at a stream queue (not taking into account wrap around).

FIG. **5** is a flow chart showing the procedure for computing a timestamp value when a cell departs from a stream queue.

FIG. **6** is a flow chart showing the procedure for checking for the idle state of a stream queue during each cell time.

FIG. **7** is a flow chart showing the procedure for computing a timestamp value when a cell arrives at a stream queue, taking into account wrap around.

FIG. **8** is a diagram of a single priority fine grain timewheel.

FIG. **9** is a diagram of a single priority coarse grain timewheel.

FIG. **10** is a diagram of a single priority ready list.

FIG. **11** is a diagram of a multi-priority fine grain timewheel.

FIG. **12** is a diagram of a multi-priority coarse grain timewheel.

FIG. **13** is a diagram of a multi-priority ready list.

FIG. **14** shows the procedure for attaching a stream queue identifier to a ready list.

FIG. **15** shows the procedure for inserting a stream queue identifier on a timewheel.

FIG. **16** shows the procedure for extracting a stream queue identifier from a ready list.

FIG. **17** shows the procedure for cell arrival timestamp computation combining scheduling and UPC shaping.

FIG. **18** shows the procedure for cell departure timestamp computation combining scheduling and UPC shaping.

FIG. **19** is a diagram of a multi-priority fine grain timewheel with one priority level per time-bin.

FIG. **20** is a diagram of a multi-priority coarse grain timewheel with one priority level per time-bin.

FIG. **21** is a diagram of timewheel time-bins and a ready lists associated with FIGS. **19** and **20**.

FIGS. **22** and **23** show timewheel scheduling operations.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In an ATM switch or multiplexer, cells arrive at a bottleneck point and are stored in buffers to await transmission through the bottleneck towards their destinations. FIG. **1** depicts the main components of an ATM buffer module serving a switch input or output port: a queue manager **2**, a scheduler **3**, a cell memory **1** and a control memory **4**. The module may, for example, be an output module or an input module of a switch.

Queue manager **2** stores arriving cells in cell memory **1** in the form of stream queues, Q_1, Q_2, . . . Q_K. Control information for each queue is stored in control memory **4**. Rather than store cells, queue manager **2** may drop cells if congestion arises. For example, a threshold-based cell discard mechanism may be used. During each cell time, queue manager **2** may choose a cell in memory to be transmitted to the next stage in the switch.

The choice of the next cell to transmit is determined by scheduler **3**, which is the focus of the present invention. In the configuration of FIG. 1, scheduler **3** interacts with queue manager **2** as follows. During each cell time, queue manager **2** queries scheduler **3**. Scheduler **3** responds with either a queue identifier or a null value. If scheduler **3** supplies a valid queue identifier, queue manager **2** removes the head-of-line cell at the corresponding stream queue in cell memory 1 and transmits the cell to the next stage.

Both queue manager **2** and scheduler **3** have access to control memory 4. Control memory 4 stores information, corresponding to each stream queue, which is used to perform buffer management and scheduling. K represents the total number of stream queues and Q_i denote the ith stream queue. Control memory **4** contains a count of the number of cells in each stream queue and other control information that may be used by queue manager 2 or scheduler 3. Scheduler **3** performs time-based scheduling. As such, a timestamp value, TS_i, is maintained for Q_i. The timestamp value represents the next time epoch at which a stream queue is eligible to be served. Also, Q_i is associated with two rates: a static, minimum guaranteed rate, M_i, and a dynamic rate, R_i, that is updated in accordance with DRC scheduling.

Scheduler 3 determines the stream queue (if any) to be served in the current cell transmission time. For a work-conserving scheduler, only the sequence of cell transmissions is important; i.e., whenever there is at least one cell in the buffers, a cell will be transmitted during the current cell time. By contrast, a non-work-conserving scheduler may allow a transmission slot to go idle even if there are cells in the buffer. In this case, the absolute time at which cells are transmitted is important.

In general, dynamic rate control (DRC) scheduling is non-work-conserving. Cells are scheduled for transmission at absolute time epochs. When cells arrive, they are queued on a per-stream basis. That is, cells corresponding to stream i are buffered in a First-In First-Out (FIFO) stream queue which is denoted as Q_i. Associated with stream queue Q_i is a rate, R_i, which is computed dynamically based on congestion information at the bottleneck points through which stream i passes. Cell scheduling is achieved by peak rate shaping each stream according to its associated dynamic rate. This can be performed by means of a timestamp value, TS_i, which is updated to reflect the next time epoch at which queue Q_i is eligible to be served. A timewheel data structure is used to store identifiers of stream queues waiting for their timestamp values to expire.

FIG. **2** is a block diagram of one embodiment of the scheduler architecture. Control memory **4** stores per queue information such as the timestamp value, rate and size of each stream queue. Rate computation unit **8** computes the rate for each stream queue based on external rate information and information stored in control memory **4**. Timestamp computation unit **7** calculates the timestamp value for each queue. Stream queues are scheduled by means of scheduling memory **5A**, which assumes the form of a timewheel data structure. Ready List **9A** contains a prioritized list of stream queues to be serviced. The ready list is explained in more detail later in the specification. Timestamp computation unit **7**, scheduling memory **5A** and ready list **9A** are all controlled and coordinated by scheduler logic unit **6**.

FIG. **3** is a block diagram of another embodiment of the scheduler architecture. The difference between this embodiment and the embodiment shown in FIG. 2 is that a second scheduling memory **5B** and a plurality of ready lists **9B**, **9C** and **9D** are used. In this architecture, the timewheel structure in one of the scheduling memories is a fine grain timewheel and the timewheel structure in the other scheduling memory is a coarse grain timewheel. These two different timewheel structures and plurality of ready lists are explained in more detail later in the specification.

SCHEDULING VIA TRAFFIC SHAPING

In an ATM network, a traffic shaper takes an input cell stream and introduces delays to certain cells, where necessary, to produce an output cell stream which conforms to the parameters of the shaping algorithm. The simplest example of a shaper is a peak rate shaper which ensures that the minimum inter-cell spacing is 1/R [seconds], where R is the specified peak rate. Traffic shaping is performed on the user side, prior to entry of the cell stream to the network. The purpose of traffic shaping is to smooth out a cell stream such that it requires less network resources and therefore, incurs a lower cost to the user.

The scheduler architecture and method of this invention is based on peak rate shaping each stream to a locally computed scheduling rate. Various forms of traffic shaping can be achieved by changing the shaping algorithm. The special case of peak rate traffic shaping will be described because it is the type of shaping required in the DRC scheduler. The peak rate shaping algorithm is simple in principle; however, a practical implementation must take into account the occurrence of wrap around due to the finite number of bits used to represent TS_i and the current time (CT). A peak rate shaping algorithm, assuming that wrap around does not occur, is described in the following section. The Wrap Around Mechanism section describes a modified algorithm to handle wrap around.

Peak Rate Shaping

In the general case, a timestamp value, TS_i, is maintained for the ith stream. The value of TS_i is updated when certain events occur, i.e., cell arrival or cell departure for stream i. Arriving stream i cells are stored in stream queue Q_i. In a given cell time, after an update of TS_i (if any) the value of TS_i represents the time at which the head-of-line cell in Q_i (if any) is eligible to be transmitted. That is, when the value of CT equals or exceeds the value of TS_i, the head-of-line cell in Q_i is eligible to be transmitted.

Initially, TS_i is set to zero and each update of TS_i increases it by a positive quantity. A current time variable, CT, keeps track of the real-time clock. Initially, CT is set to zero and

is increased by one at the beginning of each successive cell time. Assuming that TS_i and CT are each represented by n bits, after 2^n cell times, CT wraps around to the value zero. After a sufficient number of update events, the value of TS_i also wraps around. The issue of wrap around is discussed in the next section. For the following discussion, it is assumed that wrap around is not a problem.

The timestamp value TS_i is updated when one of the following events occurs:

1. Cell arrival from stream i or
2. Cell departure from stream i.

As an example of the implementation of the algorithm, assume that stream i is to be shaped to a peak rate R_i. This means that the inter-cell spacing of cell departures for stream i must be larger than, or equal to, $1/R_i$. If a stream i arrives to an empty Q_i, the cell is eligible to go out at the earlier of two times:

1. At time CT, i.e., immediately or
2. $1/R_i$ cell times after the last cell departure from Q_i.

Accordingly, the timestamp value computation for peak rate shaping upon cell arrival and departure events for stream i is shown in FIGS. **4** and **5** and is described below.

After a cell arrives (step **100**), the cell is appended to Q_i if the stream queue is not empty (step **140**). However, if the queue is empty, the stream must be scheduled (step **110**). The timestamp value TS_i is set at the maximum of (TS_i+1/R_i) or CT (step **120**). Stream i is then scheduled at time TS_i (step **130**).

After a cell is transmitted from Q_i (step **200**), no timestamp calculation is performed if the queue is empty (step **240**). However, if the queue is not empty, the stream must be scheduled (step **210**). The timestamp value TS_i is set at TS_i+1/R_i (step **220**). Stream i is then scheduled at time TS_i (step **230**). Scheduling a stream i at time TS_i means to append a stream queue identifier for the stream to the timewheel at the time-bin corresponding to time TS_i.

Wrap Around Mechanism

As an example of the wrap around mechanism, assume that TS and CT are stored using n bits. The counter CT is initialized to zero and is increased by one during each cell time. After a cycle period of 2^n cell times, CT wraps around back to the value zero. Similarly, the timestamp value TS wraps around after it is increased beyond the value 2^n-1. If CT advances past TS and wraps around, CT is said to be one cycle period ahead of TS.

Conversely, when a timestamp update event occurs, TS could be advanced past CT into the next cycle period. To keep track of the relative cycles in which the timestamp and current time lie, two 2-bit zone indicators are introduced, denoted by z_{CT} and z_{TSi} which correspond to CT and TS_i, respectively. When CT wraps around, z_{CT} is increased by one (modulo four). Similarly, when TS_i wraps around, z_{TSi} is increased by one (modulo four). The zone bits are merely two bit extensions of the registers for TS_i and CT. The interpretations of the zone bit values are shown in Table 1.

TABLE 1

Interpretation of zone indicators

Zone comparison	Interpretation
$Z_{CT} = Z_{TSi}$	CT and TS_i are in same cycle
$Z_{CT} = (Z_{TSi} - 1) \bmod 4$	CT is one cycle behind TS
$Z_{CT} = (Z_{TSi} + 1) \bmod 4$	CT is one cycle ahead of TS
$Z_{CT} = (Z_{TSi} + 2) \bmod 4$	CT is two cycles ahead of TS

In this example, it is assumed that $1/R_i<2^n$ for all streams i. This ensures that CT will never fall behind TS by more

than one cycle. A mechanism to ensure that CT will never run ahead of TS by more than one cycle will now be described. Let I_i be an idle bit for stream i, initially set to zero. If the value of I_i equals zero, the stream is considered active; otherwise, if I_i equals one, the stream is considered idle. A stream is considered idle at time CT if the most recent cell departure occurred more than $1/R_i$ cell times in the past.

Next, an independent process is introduced that cycles through all of the streams to determine which ones are idle. For those streams i that are determined to be idle, the idle bit I_i is set to one. Let N_s denote the total number of streams. It is assumed that only one queue can be tested for idleness during one cell time. To ensure that CT never advances two or more cycles ahead of TS_i, the maximum number of streams that can be supported should be less than 2^n. During each cell time, the check for idleness proceeds as shown in FIG. **6** and as described below.

At cell time $i=(i+1) \bmod N_s$ (step **300**), a determination is made whether the stream is not idle ($I_i=0$) and the queue is empty (step **310**). If both conditions are not met, the idle bit I_i is set to, or kept at, 0 (step **340**). If both conditions are met, the zones indications are analyzed as follows (step **320**): If ($z_{CT}=z_{TSi}$ and $CT-TS_i>1/R_i$) or [$z_{Ci}=(z_{TSi}+1) \bmod 4$ and $CT-TS_i+2^n<1/R_i$)], the stream is considered to be idle and I_i is set to 1 (step **330**). If both conditions are not met, I_i is set to, or kept at, 0 (step **340**).

I_i must be reset to zero whenever a stream i cell arrives. Besides this modification, the shaping algorithm takes into account the values of the zone indicators in comparing the values of CT and TS_i. The procedure for handling a cell arrival for stream i is shown in FIG. **7** and is described below.

After a cell arrives (step **400**), I_i is set to 0 and TS_i is set to TS_i+1/R_i (step **410**) and the status of the queue is checked (step **420**). If the queue is not empty, the cell is appended to the queue (step **460**). If the queue is empty, a determination is made regarding the idle state of the queue and the zone indications as follows: If $I_i=1$ or ($z_{CT}=z_{TSi}$ and $CT \geqq TS_i$) or [$z_{CT}=(z_{TSi}+1) \bmod 4$ or $z_{CT}=(z_{TSi}+2)$] mod **4** (step **430**), TS is set at CT (step **440**). If the conditions are not met, the stream is scheduled at time TS_i (step **450**).

SCHEDULING MEMORY

Timewheel

Timestamp-based algorithms for traffic shaping were discussed in the Scheduling via Traffic Shaping section above. Each stream queue Q_i has an associated timestamp value, TS_i, which indicates the time epoch when the stream queue becomes eligible for service.

During the current cell time, CT, any stream queue with a timestamp value satisfying $TS_i<CT$ is eligible for service. Although multiple stream queues may become eligible in the same time slot, only one cell from one of the queues may be transmitted in each time slot.

Therefore, a ready list of eligible stream queues is maintained. At time CT, any newly eligible stream queues are moved to the ready list. During the cell time, one of the stream queues from the ready list is chosen for service. Queue manager **2** handles the removal of the head-of-line cell from the cell memory and the transmission of the cell to the next switching stage.

The basic mechanism behind traffic shaping is simple. Cells arriving from a given stream are queued in FIFO order per stream. During a given time, the head-of-line cell in a stream queue, say Q_i, is scheduled to be transmitted at the time epoch indicated by the timestamp value, TS_i. As discussed in the previous section, the timestamp value is updated either upon arrival or departure of a cell from stream

queue Q_i. The timestamp TS_i is updated based on the current value of TS_i, the current time CT, and the dynamic rate R_i.

If the updated value of $TS_i<CT$, the head-of-line cell in stream queue Q_i is eligible to be transmitted immediately, i.e., in the current cell time. However, there may be several streams i for which $CT \geqq TS_i$.

Therefore, a ready list of eligible stream queues which have not yet been served is maintained. If the updated value of TS_i is greater than CT, then the stream queue is eligible at some future time. A timewheel structure, also called a calendar queue, is used to schedule stream queues which will become eligible for service at a future time.

The structure of the timewheel can be described as a circular array of entries numbered **0**, **1**, . . . N−1, where the nth entry points to a (possibly empty) list of eligible stream queues scheduled for time n (modulo N). After each clock tick, the value of CT is updated to point to the next entry on the timewheel. All stream queues on the list corresponding to this entry then become eligible for service. This list is then appended onto the ready list. During each cell time, one or more stream queues from the ready list are served. The maximum number of stream queues which can be served within one cell time is constrained by the speed of the logic and memory.

Reduction of Timewheel Size

The traffic shaper should be capable of supporting a wide range of rates. To support connection rates in the range 4 Kbps to 622 Mbps requires about 150 K entries in the timewheel. Each entry consists of six pairs of head/tail pointers. Assuming that up to 64 K streams are to be supported, each pointer requires 16 bits, or 2 bytes. Thus, the memory required for each entry is 24 bytes. The total memory requirement for the timewheel alone would then be 3.6 Mbytes.

This memory requirement can be reduced significantly by using two timewheels as follows (see FIGS. 8 and 9):

1. a fine grain (FG) timewheel, where each entry corresponds to one cell time.
2. a coarse grain (CG) timewheel, where each entry corresponds to a several cell times.

In this example, it is assumed that each timewheel consists of 2 K entries and stream queues are assigned to either the FG timewheel or the CG timewheel, according to rate.

With a line rate of 600 Mbps, the lowest rate for a flow that can be supported by the FG timewheel is:

$$(600\times10^6)/(2\times10^3)=300\times10^3,$$

or 300 Kbps. On the other hand, if the CG timewheel is to support a rate of 4 Kbps, then the smallest granularity that can be supported corresponds to the rate:

$$(4\times10^3)\times(2\times10^3)=8\times10^6,$$

or 8 Mbps. In this case, one entry on the CG timewheel corresponds to:

$$(600\times10^6)/(8\times10^6)=75$$

entries on the FG timewheel. To simplify things, this number is rounded to the nearest power of two, therefore, the granularity of the CG timewheel is set at 64 cell times. Then, each entry in the CG timewheel is also set at 64 entries of the FG timewheel. In units of time, the granularity of the CG timewheel is 44.8 μs compared to 700 ns for the FG timewheel. Rates are assigned to the two timewheels as follows:

FG timewheel: 300 Kbps to 600 Mbps,

CG timewheel: 4 Kbps to 300 Kbps.

In this example, for a 300 Kbps constant bit rate stream, the error introduced by the CG timewheel, as a percentage of the inter-cell distance, is approximately 3.2%.

There is no need to assign stream queues to the two timewheels in a static manner based on rate. Instead, the stream is scheduled based on the bit rate stream and the error introduced by the CG timewheel, as a percentage according to the relative values of the timestamp value TS and the value of the current time CT, as follows:

```
if TS≦CT, then
    Assign the stream element directly to the ready list.
else if TS−CT>2000, or TS is a multiple of 64, then
    Assign the stream element to the CG timewheel.
else
    Assign the stream element to the FG timewheel.
end if
```

Note that in the above pseudo-code, a stream is scheduled for the CG timewheel if the timestamp is a multiple of 64. Doing this avoids the need to access both timewheels in the same cell time.

Memory Requirement

Each entry in one of the two timewheels consists of six pairs of head/tail pointers (hp/tp), requiring 24 bytes of memory. Counting both timewheels, with 2000 entries each, the total memory requirement is then about 96 Kbytes, an order of magnitude improvement from using a single timewheel. What is lost in going from the single large timewheel to two small timewheels is coarser granularity in scheduling low rate connections and an increased probability of bunching at scheduled time slots on the coarse timewheel. However, since low rate connections generally have greater tolerance for cell delay variation, this effect is not significant.

The bunching effect due to coarser scheduling granularity can be improved by increasing the number of entries in each timewheel. For example, if the number of entries in each timewheel is doubled to 4000, the FG timewheel can support rates in the range 150 Kbps to 600 Mbps. Furthermore, the granularity of the CG timewheel is improved to 22.4 μs. In this case, each entry of the CG timewheel corresponds to 32 entries of the FG timewheel (i.e., 32 cell times).

Priority Levels

During one cell time, a fixed number of stream queues (i.e., one or two) can be served within one cell time (the maximum number depends on the memory technology that is used). However, several stream queues may become eligible during the same time slot. Thus, a backlog of eligible stream queues could form. To accommodate stream queues with different tolerances for cell delay variation (CDV), four priority levels are provided. The priorities are listed from high to low as follows:

0 Dynamic high priority (HP),

1 Real-time, short CDV (RT-S),

2 Real-time, long CDV (RT-L), and

3 Non-real-time (NRT).

HP is a dynamic assignment. Eligible stream queues that have been scheduled at their minimum guaranteed rates are automatically assigned as HP. This ensures that all stream queues receive their minimum rate guarantees on a short time-scale. The remaining three priority levels are assigned statically, according to traffic class and tolerance for cell delay variation. Streams classified as RT-S are real-time streams which have small CDV tolerances, while RT-L streams have larger CDV tolerances. Non-real-time (NRT) streams generally do not have requirements on CDV.

In general, low bit-rate real-time streams would be classified as RT-L, while high bit-rate real-time streams would be classified as RT-S. However, the CDV tolerance of a stream need not be directly related to its bit-rate. The static priority levels protect streams with small CDV tolerance from the bunching effects of streams with larger CDV tolerances. For example, consider a scenario in which there are one thousand 64 kbps voice streams sharing a 150 Mbps link with a single 75 Mbps multimedia stream.

Assuming that the multimedia stream is a constant bit rate (CBR), it needs to send a cell once every two cell times. If cells from the voice streams are bunched together at or near the same time slot, a natural consequence of superposition, the multimedia stream will suffer from severe CDV, relative to its inter-cell gap of one cell time. In the worst-case, two cells from the multimedia stream could be separated by up to one thousand voice cells.

External Storage

The scheduler data structures for a single priority level are depicted in FIGS. 8–10. For multiple priority levels, the timewheel structures and the ready list are replicated for each level (see FIGS. 11–13). For example, if there are L priority levels, then each time-bin would consist of L distinct lists, one for each priority level. A timewheel consists of a set of consecutive time-bins labelled in increasing order of time. In this embodiment, a timewheel consists of 2K time-bins, numbered consecutively from 0 to 2K−1. Note that the number of time-bins may vary depending on the particular application.

To economically handle a large range of bit rates, a plurality of timewheels are used. In this embodiment two timewheels are used: a fine grain and a coarse grain timewheel. The fine grain timewheel time-bins correspond to cell times numbered **0**, **1**, . . . 2K−1. The coarse grain timewheel time-bins correspond to cell times numbered **0**, **64**, **128**, . . . , (64*2K)−1. Note that the different timewheels do not have to contain the same number of time-bins. Generally speaking, the fine grain timewheel is used for scheduling high rate streams, while the coarse grain timewheel is used for scheduling lower rate streams, although this distinction is not a strict property of the scheduling algorithms to be described below.

Each timewheel time-bin is associated with stream queues which are scheduled for the same time slot. Because up to 64K streams are to be supported, a stream pointer, or stream queue identifier, is identified with a 16-bit word. Each timewheel time-bin consists of head and tail pointers (hp and tp) which point to locations in a stream pointer memory. These pointers for a list for each time-bin. The stream pointer memory consists of 64K entries. Each entry in the stream pointer memory is a 16-bit pointer to another stream. Thus, the stream pointer memory is logically 64K deep and 16 bits wide. The stream pointer memory is defined as follows:

Word V[0 . . . (64K−1)],

where Word denotes a 16-bit integer type. The coarse timewheels are defined by:

Queue T_C [0 . . . 3][0 . . . (2K−1)],

while the fine timewheels are defined by

Queue T_F [0 . . . 3][0 . . . (2K−1)],

where the type Queue is a compound data type defined as follows:

Word hp;
Word tp;

For example, the head pointer at time 2 on the coarse grain timewheel and on priority 3 is denoted by:

T_C[3][2].hp

Both the stream pointer memory and the timewheel memory are external to the scheduler control logic.

The size of the ready list is a measure of the backlog in the scheduler. By applying local dynamic rate control (DRC), the scheduler can be made nearly work-conserving. Since the DRC computation is based on queue length information, it is necessary to maintain a count of the number of entries on the ready list. This can be done by storing a count of the number of stream queue identifiers in each time-bin. Since there are at most 64 K active streams, the maximum number of stream queue identifiers in one time-bin is 64 K, so the counter size needed is at most 16 bits. Therefore, the counts for the coarse and fine timewheels are defined, respectively, as follows:

Word M_C [0 . . . (2K−1)]

Word M_F [0 . . . (2K−1)]

When the current time value, CT, advances to point to the next time-bin, all stream queue identifiers associated with the time-bin become eligible for service. That is, the scheduled timestamps for the streams corresponding to these stream queue identifiers expire and the streams are ready to be served. Expired stream queue identifiers are maintained in a ready list. The ready list contains stream queue identifiers which are ready to be served, but which have not yet been processed. When the scheduler receives an external stream service request, a stream queue identifier is removed from the head of the ready list and the stream queue identifier is either sent to an internal output queue or transmitted to an external process.

Within the control logic there are bit maps which are in one-to-one correspondence with the timewheel memories:

Bit B_C [0 . . . 3][0 . . . 2K−1]

Bit B_F [0 . . . 3][0 . . . 2K−1]

B_C and B_F, respectively, denote the coarse and fine grain bit maps. The bit maps are initialized to zero, indicating that all timewheel time-bins are initially empty. A value of one in a bit map entry indicates that the corresponding timewheel time-bin is not empty. There is one ready list for each priority level:

Queue r[0 . . . 3]

An empty ready list i is indicated by setting r[i].hp=0. We also define an integer variable rc which counts the total number of entries on all ready lists.

Scheduling

Current time is stored in a 17-bit counter denoted CT. Two auxiliary bits stored in z_{CT} indicate the zone of CT. The zone, z_{CT}, takes on the values 0, 1, 2, or 3. When CT wraps around from 128K−1 to zero, z_{CT} is incremented by one (modulo four). Similarly, the timestamp value, TS_i, for stream queue Q_i is stored as a 17-bit number with the zone indicated by two bits stored in z_{TSi}. If $z_{CT}=z_{TSi}$, i.e., if current time and the timestamp value are in the same zone, then CT and TS_i can be compared directly. If $z_{CT}=(z_{TSi}-1)$ mod 4, then Ts_i represents a time in the next zone, i.e., in the future with respect to CT. Otherwise, if $z_{CT}=(z_{TSi}+1)$ mod 4

or $z_{CT}=(Z_{TS_i}+2)$ mod 4, then TS_i represents a time in a previous zone, i.e., in the past with respect to CT.

After each cell time, the current time CT is advanced by one. Before CT is advanced, any stream queue identifiers associated with the time-bin at CT must be attached to the appropriate ready lists. FIG. **14** describes the procedure for attaching stream queue identifiers to the appropriate ready list. The first part of the procedure determines whether CT corresponds to the coarse grain or the fine grain timewheel. All time-bins which are located at multiples of 64 are stored on the coarse grain timewheel.

The counter memory M_X is read once. The timewheel head and tail pointers are read once for each priority level. This gives a total of eight reads to the timewheel. The stream pointer memory, V, is written at most once for each priority level, giving a total of four reads. The internal bit map B_X is accessed at most twice for each priority level, giving a total of eight accesses. The ready list pointer r is written four times and read twice for each priority level, for a total of twelve accesses. Finally, a read-modify-write access is needed to increment rc. If separate memories are used for the three external memories, the worst-case number of memory access times required for this operation is eight. A summary of the accesses to memory for transferring the lists from the time-bin corresponding to current time CT to the corresponding ready lists are summarized in Table 2.

TABLE 2

Accesses to memory			
Memory	Read	Write	Read-Modify-Write
M_x	1	0	0
T_x	8	0	0
V	0	4	0
B_x	4	4	0
r	4	8	0
rc	0	0	1

If a stream queue identifier is to be added to the timewheel at position TS and that stream is to be scheduled at priority i, the procedure described in FIG. **15** determines the timewheel (coarse or fine) and time-bin at which the stream queue identifier should be inserted. The variable X is set to F if the fine grain timewheel is to be used and C if the coarse grain timewheel is to be used. The time-bin location is stored as variable t.

This procedure requires one read-modify-write to update the count M_X. In the worst case, two writes to T_X are needed, and one write to V. One write and one read access are made to the internal bit map B_X. Therefore, in the worst case, two external memory accesses are needed to insert a new stream queue identifier to the timewheel. The accesses to memory for inserting a new stream queue identifier to the timewheel are summarized in Table 3.

TABLE 3

Accesses to memories for inserting new stream queue identifiers to the timewheel			
Memory	Read	Write	Read-Modify-Write
M_x	0	0	1
T_x	0	2	0
V	0	1	0
B_x	1	1	0

FIG. **16** describes the procedure for extracting a stream queue identifier from the ready lists in order of priority. When the ready list for priority **0** (high) is exhausted, the ready list for priority **1** (lower) is examined, etc. The extracted stream queue identifier is stored as variable q, which is passed on to another process which transmits the head-of-line cell from the queue corresponding to q.

For each stream queue identifier that is extracted from the ready list, at most one read from the stream pointer memory V is required. Two reads and one write to the ready list pointers r are needed. Finally, a read-modify-write operation is necessary to increase the counter ready list counter rc. The accesses to memory for inserting a new stream queue identifier to a timewheel are summarized in Table 4.

TABLE 4

Accesses to memories for inserting new stream queue identifier to timewheel			
Memory	Read	Write	Read-Modify-Write
V	0	1	0
r	2	1	0
rc	0	0	1

Accesses in One Cell Time

The operations required in one cell time are listed (in order) in Table 5. In the worst case, 17 memory accesses (in parallel for separate memories) are required during one cell time. This number could be improved by reading and writing head/tail pointers on the timewheel at the same time. Note that if time allows during a cell time, step **4** may be repeated. Table 5. Memory accesses for inserting new stream queue identifier to timewheel.

TABLE 5

Memory accesses for inserting new stream queue identifier to timewheel.	
Cell operation	Worst-case Accesses
1. Reschedule (insert new stream)	2
2. Schedule (insert new stream)	2
3. Transfer list of stream queue identifiers from CT-bin to ready list	12
4. Extract stream queue identifier from ready list	1
	17

To summarize, the method for scheduling stream queues containing cells in an ATM switch, without taking into account priority, comprises the following main steps:

(a) calculating a scheduling rate value for each stream;

(b) calculating a timestamp value for each stream queue based on its scheduling rate value;

(c) scheduling each stream queue by assigning a stream queue identifier to a first timewheel scheduling memory time-bin based on its timestamp value;

(d) transferring a list of stream queue identifiers from a time-bin on the timewheel to a ready list when a current time value equals the time-bin value;

(e) choosing a first stream queue identifier from the ready list; and

(f) transmitting a first cell in the stream queue corresponding to the chosen stream queue identifier;

wherein the timestamp and current time values cycle.

The method for scheduling stream queues containing cells in an ATM switch, taking into account priority, comprises the following main steps:

(a) calculating a scheduling rate value for each stream;

(b) calculating a timestamp value for each stream queue based on its scheduling rate value;

(c) assigning one of at least two priority levels to each stream queue, wherein the priority levels are assigned different values from high to low;

(d) scheduling each stream queue by assigning a stream queue identifier to a timewheel scheduling memory time-bin based on its timestamp value and its priority level;

(e) transferring a list of stream queue identifiers from a time-bin on the timewheel to a ready list at the appropriate priority level when a current time value equals the time-bin value;

(f) choosing a first stream queue identifier from the highest priority non-empty ready list; and

(g) transmitting a first cell in the stream queue corresponding to the chosen stream queue identifier;

wherein the timestamp and current time values cycle.

One aspect of the multiple priority level embodiment described above is that during one cell time, the different priority lists at the time-bin corresponding to the current time value CT are transferred to the corresponding ready lists one at a time. An alternate embodiment of the scheduler architecture supporting multiple priorities is described below with reference to FIGS. 19–23.

FIGS. 19 and 20 show examples of this embodiment with two timewheels. There are L=4 priority levels and the granularity of the coarse grain timewheel is G=64. The fine grain timewheel consists of M*L entries and the coarse grain timewheel consists of N*L entries. Each time-bin on the FG timewheel corresponds to one cell time and each time-bin on the CG timewheel corresponds to G cell times. For this example, it is assumed that the values of L and G are both powers of two. The time-bins of the FG timewheel are assigned priority levels by labelling the time-bins cyclically in priority level order. For example, 0, 1, 2, 3, 0, 1, 2, 3, etc., as shown in FIG. 19. Similarly, the time-bins of the CG timewheel are assigned priority levels.

Each timewheel entry (i.e. time-bin) consists of a list of stream queue identifiers. During each cell time the list at the time-bin corresponding to current time CT is transferred to the ready list at the priority level assigned to the time-bin. The ready lists are also lists of stream queue identifiers. During each cell time, one stream queue identifier is removed from the non-empty ready list of the highest priority (if one exists). The first cell from the corresponding stream queue is transmitted and then the stream queue is rescheduled if it remains non-empty.

FIGS. 22 and 23 show the procedures of scheduling a new stream with associated timestamp value TS at priority level P. Note that these figures do not take into account possible wrap around situations. If a wrap around situation occurs, additional procedures along the lines discussed in the Wrap Around Mechanism section would be required.

The first step in scheduling a new stream at time TS and priority level P is to compare TS to CT (step 700). If TS is less than CT, the stream queue identifier is appended to the tail of the priority P ready list (step 705) and the procedure ends.

If TS is not less than CT, (TS–CT) is compared to (G*N*L) (step 710). G represents the granularity of the coarse grain timewheel. If (TS–CT) is greater than (G*N*L), TS' is set to the possible scheduled time value of CT+G*((N–1)*L+P) (step 715). N determines the size of the fine grain timewheel which is given by N*L. L represents the number of priority levels. Then P' is set to correspond to the priority of time-bin TS (Step 720). Note that if (TS–CT) is not greater than (G*N*L), TS is not modified and the procedure moves directly to step 720.

Next TS' is set to TS+(P–P') (step 725). If TS is a multiple of G and P=0 (step 735), TS' is set to TS'–L (step 740). Then TS' is compared to CT (step 745). Note that if TS is not a multiple of G or P=0, TS' is set to TS'–L, TS' is not modified and the procedure moves directly to step 745.

If TS' is less than CT, the stream queue identifier is appended to the tail of the priority P ready list (step 750) and the procedure ends. If TS' is not less than CT, TS'–CT is compared to M*L. M*L (step 747). M determines the size of the coarse grain timewheel which is given by M*L. If TS'–CT is not less than M*L, the stream is scheduled on the coarse grain timewheel as shown in FIG. 23 and as described below.

If TS'–CT is less than M*L, P' is compared to P (step 755). If P' is greater than or equal to P, the stream queue identifier is appended to the tail of the list at time-bin TS' on the fine grain timewheel (step 760) and the procedure ends.

If P' is greater than or equal to P, the stream queue identifier is inserted at the head of the list at time-bin TS' on the fine grain timewheel (step 765) and the procedure ends.

If the stream is to be scheduled on the coarse grain timewheel, the procedure in FIG. 23 is followed. First, I is set to TS'/G (step 800). Then P' is set to the priority level corresponding to time-bin G*I (step 810). Next, TS" is set to G*(I+(P–P")) (step 820).

Then P' is compared to P (step 830). If P' is greater than or equal to P, the stream queue identifier is appended to the tail of the list at time-bin TS" on the coarse grain timewheel (step 840) and the procedure ends. If P' is not greater than or equal to P, the stream queue identifier is inserted at the head of the list at time-bin TS" on the coarse grain timewheel (step 850) and the procedure ends.

RATE COMPUTATION

In DRC scheduling the scheduling rate for a given stream is updated dynamically. In DRC scheduling, the dynamic rate, R_{drc}, is computed as the sum of a minimum guaranteed rate M and an excess rate E which reflects the excess bandwidth available to the stream at various bottleneck points along its path:

$$R_{drc}=M+E.$$

A local dynamic rate, E_{loc}, can be computed based on the utilization observed as queues are transferred from the ready list. In this way, the scheduler is made nearly work-conserving at the local bottleneck. An external excess rate, E_{ext}, computed at a downstream bottleneck within the switch, may serve as input to the rate computation engine.

In this case, the DRC excess rate is taken as:

$$E=\min(E_{loc}, E_{ext})$$

The rate, E_{ext}, may itself be the minimum of several DRC rates computed at bottleneck points along the path of the stream inside the switch. This rate information is carried by internal resource management (IRM) cells.

Local DRC Rate Computation

DRC scheduling is discussed in more detail in co-pending application Ser. No. 08/924,820. A brief description of how DRC can be applied locally will be provided. The local DRC

excess rate, denoted by Elm can be calculated based on the measured local utilization, $\hat{U}_{loc}$. A proportional-derivative (PD) controller iteratively computes a new value of E_{loc} so as to minimize the difference between the measured utilization and the target utilization U_{loc}. The controller has the form:

$$E_{loc}(n+1)=E_{loc}(n)+\alpha_1(U_{loc}-\hat{U}_{loc})+\alpha_2(U_{loc}-\hat{U}_{loc})$$

where the filter coefficients α_1 and α_2 are chosen to ensure stability and fast convergence. Class-based E can be computed in analogous way.

ABR Virtual Source

In ABR virtual source control (see, "Traffic Management Specification, Version 4.0," The ATM Forum, 1996), the scheduler mimics the behavior of an ABR source. ABR resource management (RM) cells carry explicit rate (ER) information which determines the rate at which cells are transmitted. This external rate, which we denoted by R_{abr}, is used by the local scheduler to shape the ABR stream. ABR virtual source control can be easily combined with DRC scheduling by taking the scheduling rate for an ABR stream as the minimum of the rates computed for DRC and ABR VS control; i.e.,

$$R=\min(R_{abr}, R_{drc});$$

wherein, R_{drc} represents a locally computed rate for DRC scheduling. An example of an algorithm for calculation of R_{abr} is contained in the "Traffic Management Specification, Version 4.0."

Usage Parameter Control

In addition to scheduling cells for stream i at the scheduling rate R_i, the scheduler architecture can be used to perform traffic shaping for each stream in conformance with the UPC (Usage Parameter Control) specification in "Traffic Management. Specification, Version 4.0," The ATM Forum, 1996). In particular, how stream i can be simultaneously scheduled at rate R_i and shaped to conform to GCRA(1/PCR_i, 0) and GCRA(1/SCR_i, THi) is discussed briefly below. (For a specification of the Generic Cell Rate Algorithm for UPC policing, see "Traffic Management Specification, Version 4.0," The ATM Forum, 1996).

The purpose of shaping a stream is to force it to conform to UPC parameters which are policed at the next hop in the network (e.g., the interface to a separate subnetwork). This prevents cell discard or cell tagging incurred by the policer on nonconforming cells. FIG. **17** shows the procedure for computing the timestamp (not accounting for wrap-around) for combined rate scheduling and UPC shaping when a cell arrives to the stream i queue. After a cell arrives to queue i (step **500**), the status of the queue is checked (step **510**). If the queue is empty, the scheduling timestamp TS_i is updated according to TS_i=MAX[{TS_i+MAX(1/PCR_i, 1/R_i)}, CT] (step **520**).

Then the shaping timestamp TS_i' is compared with CT (step **530**). If TS_i' is less than or equal to CT, TS_i' is set to CT (step **540**). The stream is scheduled at time MAX(CT, TS_i) (step **560**), TS_i' is updated according to TS_i'=TS_i'+1/SCR_i (step **570**) and the cell is appended to queue i (step **580**).

In step (**530**), if TS_i' is greater than CT, then TS_i' is compared with CT+TH_i (step **550**). If TS_i'>CT+TH_i, the stream is scheduled at time MAX(TS_i'−TH_i, TS_i) (step **555**). Then the cell is appended to queue i.

In step (**550**), if TS_i' is less than or equal to CT+TH_i, the stream is scheduled at time MAX(CT, TS_i) (step **560**), TS_i' is updated according to TS_i'=TS_i'+ 1/SCR_i (step **570**) and then the cell is appended to queue i (step **580**).

FIG. **18** shows the procedure for computing the timestamp (not accounting for wrap-around) for combined rate scheduling and UPC shaping when a cell departs from a stream i queue. The cell is removed and transmitted from queue i (step **600**). The status of the queue i is checked (step **610**). If the queue is empty, the procedure ends (step **650**). Otherwise, the scheduling timestamp is updated according to TS_i=TS_i+MAX(1/R_i, 1/PCR_i) (step **620**). The shaping timestamp is updated according to TS_i'=TS_i'+1/SCR_i (step **630**). Then the stream is scheduled at time MAX(TS_i, TS_i'−TH_i) (step **640**), before the procedure ends (step **650**).

The present invention is a scalable and flexible architecture for implementing DRC scheduling in an ATM switch. The architecture performs peak rate shaping of streams, where the shaping rates are determined according to the DRC scheme. The scheduler is based on a timewheel data structure where stream queues await service until their computed timestamps expire. A ready list stores eligible stream queues which have not yet been served.

To achieve a wide range of rates without employing large memories, the scheduler is implemented with at least two timewheels: a fine grain timewheel and a coarse grain timewheel.

The timewheel structure is augmented with a plurality of priority levels, four in this example. The high priority level is assigned dynamically to ensure that streams will be able to meet their minimum rate guarantees. The remaining priority levels provide QoS differentiation at a coarse level. The timestamp value for each stream queue is updated, as appropriate, to achieve peak rate shaping in accordance with the rate determined by the DRC scheme.

While the above is a description of the invention in its preferred embodiments, various modifications and equivalents may be employed. Therefore, the above description and illustration should not be taken as limiting the scope of the invention which is defined by the claims.

What is claimed is:

1. An apparatus for scheduling stream queues serving cells in an ATM switch comprising:

a cell memory connected to a queue manager unit that stores ATM cells organized into stream queues; and

a control memory connected to a scheduler unit and the queue manager that stores queue information;

wherein the scheduler unit selects a stream queue to be serviced, based on the queue information in the control memory, and comprises a timewheel scheduling memory that stores stream queue identifiers in a series of time-bins; and

wherein the queue manager controls the receipt and transmission of ATM cells based on the congestion of the ATM switch and on the queue information in the control memory.

2. The apparatus of claim **1**, wherein the scheduler unit further comprises:

a rate computation unit that computes the rate for each stream queue based on external rate information and the queue information in the control memory;

a time stamp computation unit that calculates a time stamp value for each stream queue;

at least one ready list that stores the stream queue identifiers that are ready to be serviced;

a scheduler logic unit that coordinates the operation of the timewheel scheduling memory, the time stamp computation unit and the ready list.

3. The apparatus of claim **1**, wherein the scheduler unit comprises a plurality of timewheel scheduling memories,

wherein time-bins in a first timewheel scheduling memory are assigned values corresponding to one cell time and time-bins in the other timewheel scheduling memories are assigned different values corresponding to more than one cell time.

4. The apparatus of claim 3, wherein the scheduler unit further comprises:

a rate computation unit that computes the rate for each stream queue based on external rate information and then queue information in the control memory;

a time stamp computation unit that calculates a time stamp for each stream queue;

a ready list that stores the stream queue identifiers that are ready to be serviced;

a scheduler logic unit that coordinates the operation of the plurality of timewheel scheduling memories, the time stamp computation unit and the ready list.

5. The apparatus of claim 4,

wherein each time-bin consists of a plurality of lists, each list corresponding to a different priority level, and

wherein there are a plurality of ready lists, each ready list corresponding to one of the different priority levels.

6. The apparatus of claim 4,

wherein each time-bin consists of a single list, and

wherein there are a plurality of ready lists, each ready list corresponding to a different priority level.

7. A method for scheduling stream queues containing cells in an ATM switch comprising the steps of:

(a) calculating a scheduling rate value for each stream;

(b) calculating a timestamp value for each stream queue based on its scheduling rate value;

(c) scheduling each stream queue by assigning a stream queue identifier to a first timewheel scheduling memory time-bin based on its timestamp value;

(d) transferring a list of stream queue identifiers from a time-bin on the timewheel to a ready list when a current time value equals the time-bin value;

(e) choosing a first stream queue identifier from the ready list; and

(f) transmitting a first cell in the stream queue corresponding to the chosen stream queue identifier;

wherein the timestamp value and current time value cycle.

8. The method of claim 7, wherein the timestamp value of each stream queue is recalculated at the occurrence of one of at least a cell arriving at an empty stream queue and a cell departing from a non- empty stream queue.

9. The method of claim 7, wherein the current time value never falls behind the timestamp value by more than one cycle or moves ahead of the timestamp value by more than one cycle.

10. The method of claim 7, wherein in step (c), each stream queue identifier is assigned to one of a plurality of timewheel scheduling memories at a time-bin based on its timestamp value.

11. A method for scheduling stream queues containing cells in an ATM switch comprising the steps of:

(a) calculating a scheduling rate value for each stream;

(b) calculating a timestamp value for each stream queue based on its scheduling rate value;

(c) assigning one of at least two priority levels to each stream queue, wherein the priority levels are assigned different values from high to low;

(d) scheduling each stream queue by assigning a stream queue identifier to a timewheel scheduling memory time-bin based on its timestamp value and its priority level;

(e) at each priority level, transferring a list of stream queue identifiers from a time-bin on the timewheel to a ready list when a current time value equals the time-bin value;

(f) choosing a first stream queue identifier from the highest priority non-empty ready list; and

(g) transmitting a first cell in the stream queue corresponding to the chosen stream queue identifier;

wherein the timestamp value and current time value cycle.

12. The method of claim 11, wherein a new stream queue identifier is placed on the time-bin corresponding to the timestamp value and on a list in the time-bin corresponding to the priority level.

13. The method of claim 11, wherein the time-bins are assigned priorities cyclically in priority level order and a new stream queue identifier is placed on the time-bin corresponding to the timestamp value and the priority level.

14. The method of claim 11, wherein the timestamp value of each stream queue is recalculated at the occurrence of one of at least a cell arriving at an empty stream queue and a cell departing from a non-empty stream queue.

15. The method of claim 11, wherein the current time value never falls behind the timestamp value by more than one cycle or moves ahead of the timestamp value by more than one cycle.

16. The method of claim 11, wherein in step (d) each stream queue identifier is assigned to a time-bin in one of a plurality of timewheel scheduling memories based on its timestamp value.

17. The method of claim 14, wherein the scheduling rate value computed for each stream is the minimum of a locally computed rate and an external rate.

18. The method of claim 14, wherein,

the timestamp calculation is augmented to perform both scheduling of the stream based on the scheduling rate value and

shaping the stream in conformance with usage parameter control policing parameters.

19. A method of scheduling a stream composed of a sequence of cells to successively transmit each cell towards a downstream side, comprising the steps of:

calculating, on the basis of a dynamic rate on the downstream side, a scheduling timestamp representative of timing at which each cell of the stream is to be scheduled; and

deciding a shaping timestamp of each cell on the basis of the scheduling timestamp

wherein the calculating step comprises the step of:

calculating the scheduling timestamp with reference to UPC (Usage Parameter Control) parameters.

20. A method of scheduling a stream composed of a sequence of cells to successively transmit each cell towards a downstream side, comprising the steps of:

calculating a scheduling timestamp on the basis of a peak cell rate (PCR) of the stream, a sustainable cell rate (SCR), a burst threshold (TH), and a dynamic rate; and controlling a cell rate without congestion on the basis of the scheduling timestamp calculated

wherein the calculating step comprises the step of:

calculating the scheduling timestamp with reference to UPC (Usage Parameter Control) parameters.

21. A scheduler for use in scheduling a stream composed of a sequence of cells to successively transmit each cell towards a downstream side, comprising:

calculating means for calculating a scheduling timestamp with reference to a dynamic rate computed on conges-

tion on the downstream side to specify scheduling time at which each cell of the stream is to be scheduled; and

deciding means for deciding output timing of each cell in the shaped manner on the basis of the scheduling timestamp and the current time

wherein the calculating means calculates the scheduling timestamp with reference to a peak cell rate (PCR) of the stream along with the dynamic rate.

22. A scheduler as claimed in claim **21**, wherein the calculating means calculates the scheduling timestamp with reference to a sustainable cell rate (SCR) and a burst threshold (TH) for the stream together with the peak cell rate and the dynamic rate.

23. A scheduler for use in scheduling a stream composed of a sequence of cells to successively transmit each cell towards a downstream side, comprising:

calculating means for calculating a scheduling timestamp with reference to a dynamic rate computed on congestion on the downstream side to specify scheduling time at which each cell of the stream is to be scheduled; and

deciding means for deciding output timing of each cell in the shaped manner on the basis of the scheduling timestamp and the current time

wherein the calculating means comprises:

first means for calculating, on arrival of each cell in the stream, a first timestamp of each cell on the basis of the dynamic rate of the stream, a peak cell rate (PCR) of the stream, and a current time.

24. A scheduler as claimed in claim **23**, wherein the calculating means further comprises:

second means for calculating a second timestamp with reference to a sustainable cell rate (SCR) together with the first means for calculating the first timestamp.

25. A scheduler as claimed in claim **24**, wherein the second means calculates the second timestamp also with reference to a predetermined burst threshold (TH) for the SCR.

26. A scheduler as claimed in claim **25**, wherein the deciding means deciding the shaping timestamp from the first and the second timestamps.

27. A scheduler for use in scheduling a stream composed of a sequence of cells to successively transmit each cell towards a downstream side, comprising:

calculating means for calculating a scheduling timestamp with reference to a dynamic rate computed on congestion on the downstream side to specify scheduling time at which each cell of the stream is to be scheduled; and

deciding means for deciding output timing of each cell in the shaped manner on the basis of the scheduling timestamp and the current time

wherein the deciding means is operable to decide the shaping timestamp on cell departure; and

the calculating means comprises:

first means for calculating a first timestamp on the basis of a timestamp assigned to a preceding one of the cells in the stream, the dynamic rate of the stream, and a peak cell rate (PCR) of the stream, to obtain the scheduling timestamp with reference to the first timestamp.

28. A scheduler as claimed in claim **27**, wherein the calculating means further comprises:

second means for calculating a second timestamp with reference to a sustainable cell rate (SCR) and a burst threshold (TH) to obtain the scheduling timestamp.

29. A scheduler as claimed in claim **28**, wherein the calculating means comprises:

means for calculating the scheduling timestamp from the first and the second timestamps.

30. A scheduler as claimed in claim **29**, wherein the second means comprises:

comparing means for comparing the first timestamp with a resultant timestamp obtained by subtracting a threshold from the second timestamp to assign a maximum one of the first timestamp and the resultant timestamp as the scheduling timestamp.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,389,019 B1 Page 1 of 1
DATED : May 14, 2002
INVENTOR(S) : Ruixue Fan, Brian L. Mark, Gopalakrishan Ramamurthy and Alexander T. Ishii

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2,
Line 57, delete "Vol. Oct. 31" insert -- Vol. 31, Oct, --

Column 7,
Lines 28 & 29, delete "times-tamp" insert -- timestamp --;

Column 8,
Line 50, delete "<CT" insert -- ≤CT --

Column 9,
Line 3, delete "<CT" insert -- ≤CT --

Column 12,
Line 66, delete "Ts_I" insert -- TS_I --

Signed and Sealed this

Twenty-second Day of October, 2002

Attest:

JAMES E. ROGAN

Attesting Officer *Director of the United States Patent and Trademark Office*

US006366762B1

(12) United States Patent
Miller et al.

(10) Patent No.: US 6,366,762 B1
(45) Date of Patent: Apr. 2, 2002

(54) SYSTEM AND METHOD FOR MEASURING ROUND TRIP DELAY ON THE PAGING AND ACCESS CHANNELS

(75) Inventors: David S. Miller, Carlsbad, CA (US); Nadav Levanon, Ramat-Gan (IL); Avneesh Agrawal, San Jose, CA (US)

(73) Assignee: Qualcomm, Inc., San Diego, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: 09/338,163

(22) Filed: Jun. 22, 1999

Related U.S. Application Data

(60) Provisional application No. 60/127,472, filed on Mar. 31, 1999.

(51) Int. Cl.[7] H04B 17/00; H04B 7/185; G01S 3/52; G01S 5/02

(52) U.S. Cl. 455/67.6; 455/67.1; 455/12.1; 455/456; 342/357.05; 342/357.08; 342/351.1; 342/418

(58) Field of Search 342/352, 357.05–357.06, 342/357.08–357.09, 351.1, 418, 457, 375; 455/67.6, 67.1, 456, 12.1

(56) References Cited

U.S. PATENT DOCUMENTS

5,280,629 A * 1/1994 Lo Balbo et al. 455/67.6
5,515,062 A * 5/1996 Maine et al. 342/457
5,943,606 A * 8/1999 Kremm et al. 455/12.1
6,047,161 A * 4/2000 Sowles et al. 455/12.1
6,107,959 A * 8/2000 Levanon 342/357
6,137,441 A * 10/2000 Dai et al. 342/357.05
6,298,238 B1 * 10/2001 Dai 355/456

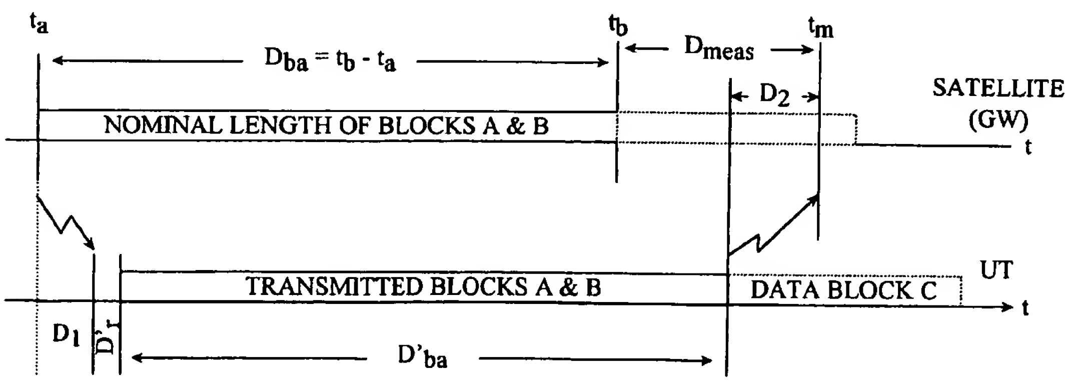
* cited by examiner

Primary Examiner—Dwayne Bost
Assistant Examiner—Raymond B. Persino
(74) Attorney, Agent, or Firm—Philip R. Wadsworth; Gregory D. Ogrod

(57) ABSTRACT

A system and method for determining a round trip delay of signals transmitted between first and second objects, such as a satellite and a mobile telephone, that move relative to each other. A first signal is transmitted from the first object to the second object. The first signal is received at the second object after a propagation delay D_1, the delay D_1 being the time taken by the first signal to traverse from the first object to the second object. A frequency of the first signal is measured at the second object. The second object then transmits to the first object a second signal containing a report of the measured first frequency. The second signal is received at the first object after a propagation delay D_2, D_2 being the time taken by the second signal to traverse from the second object to the first object. The first object measures a frequency of the second signal. The first object then determines the range between the first and second object from the measured overall delay and the measured frequencies of the first and second signals.

36 Claims, 5 Drawing Sheets



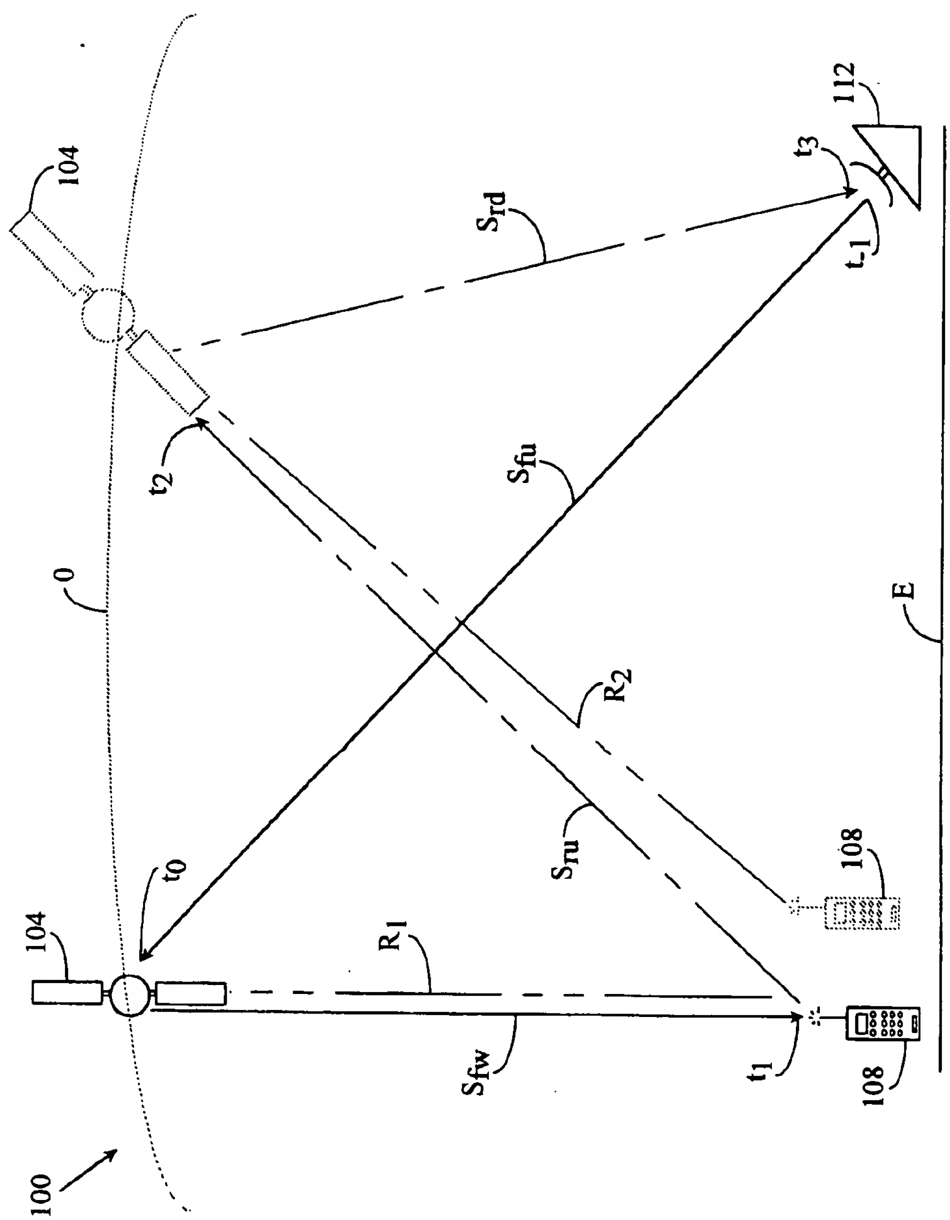


FIG. 1

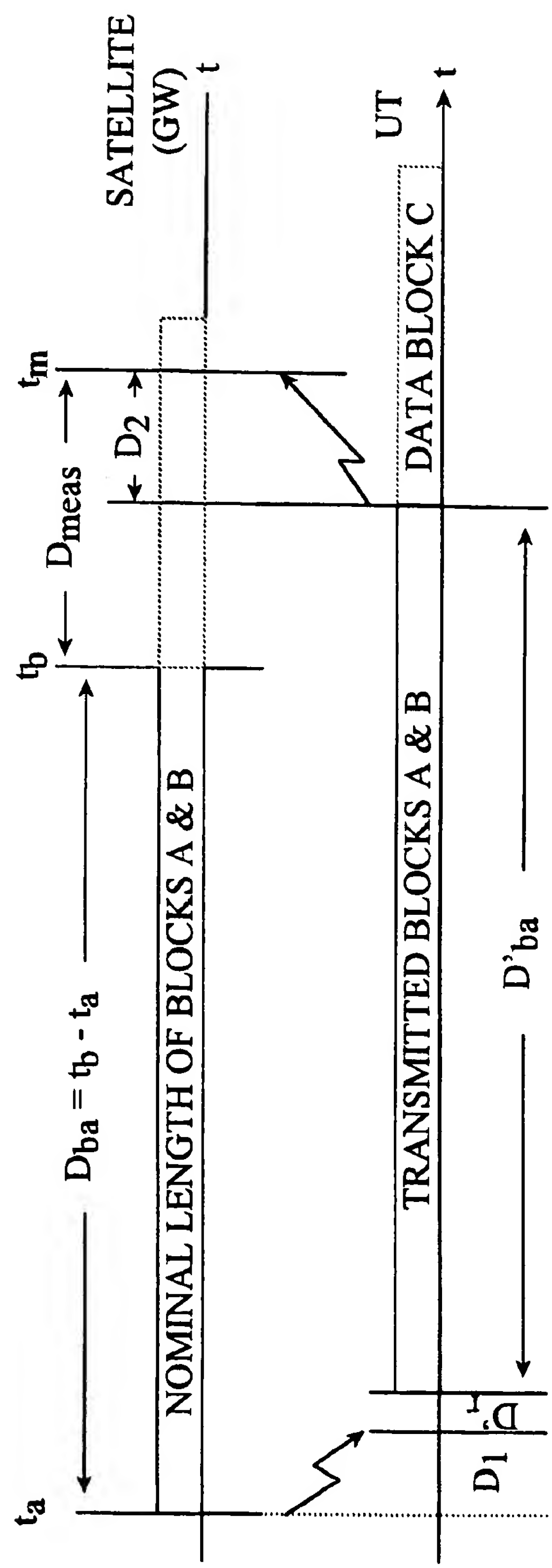

SATELLITE (GW)
t
UT
t
DATA BLOCK C
t_m
D_2
D_{meas}
t_b
NOMINAL LENGTH OF BLOCKS A & B
TRANSMITTED BLOCKS A & B
$D_{ba} = t_b - t_a$
D'_{ba}
D_1
D'_1
t_a

FIG. 2

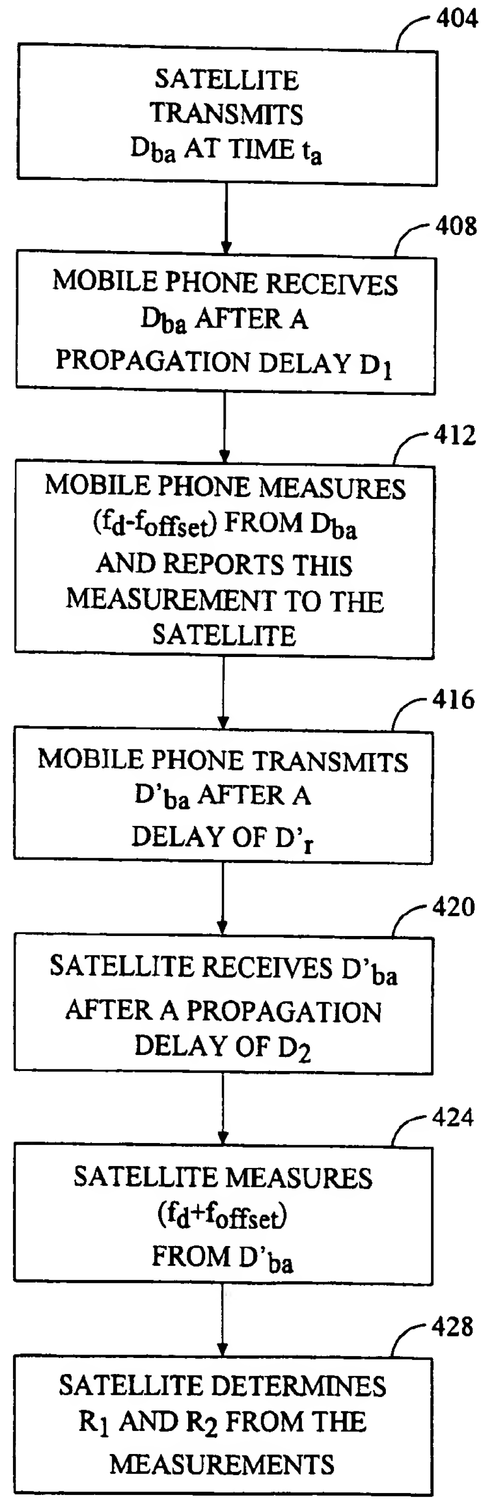

404
SATELLITE TRANSMITS D_{ba} AT TIME t_a
408
MOBILE PHONE RECEIVES D_{ba} AFTER A PROPAGATION DELAY D_1
412
MOBILE PHONE MEASURES (f_d-f_{offset}) FROM D_{ba} AND REPORTS THIS MEASUREMENT TO THE SATELLITE
416
MOBILE PHONE TRANSMITS D'_{ba} AFTER A DELAY OF D'_r
420
SATELLITE RECEIVES D'_{ba} AFTER A PROPAGATION DELAY OF D_2
424
SATELLITE MEASURES (f_d+f_{offset}) FROM D'_{ba}
428
SATELLITE DETERMINES R_1 AND R_2 FROM THE MEASUREMENTS

FIG. 3

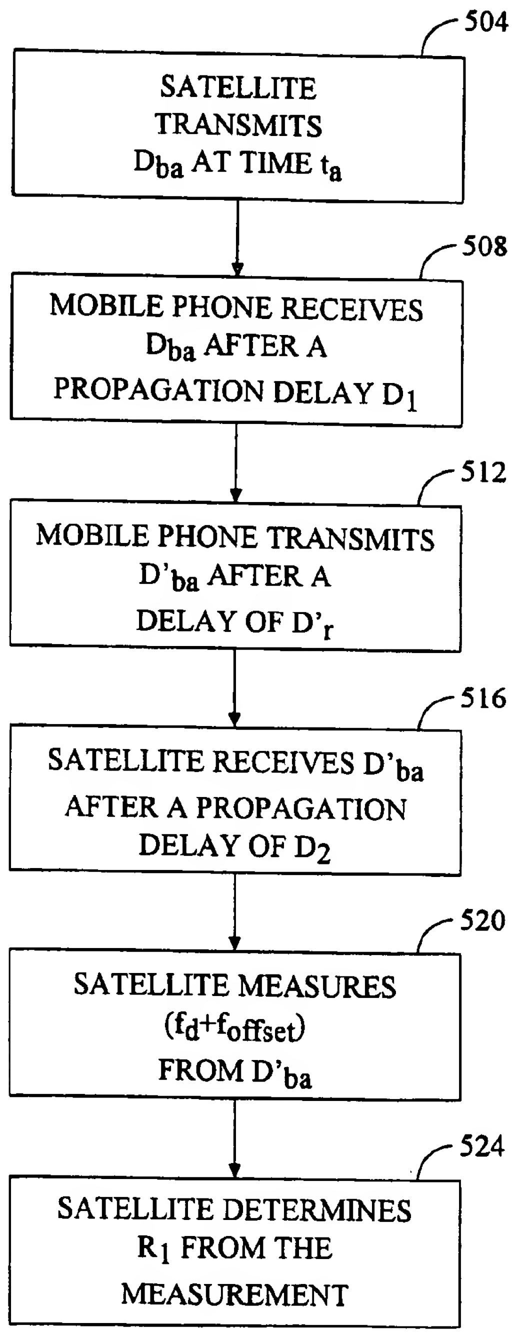

504
SATELLITE TRANSMITS D_{ba} AT TIME t_a
508
MOBILE PHONE RECEIVES D_{ba} AFTER A PROPAGATION DELAY D_1
512
MOBILE PHONE TRANSMITS D'_{ba} AFTER A DELAY OF D'_r
516
SATELLITE RECEIVES D'_{ba} AFTER A PROPAGATION DELAY OF D_2
520
SATELLITE MEASURES (f_d+f_{offset}) FROM D'_{ba}
524
SATELLITE DETERMINES R_1 FROM THE MEASUREMENT

FIG. 4

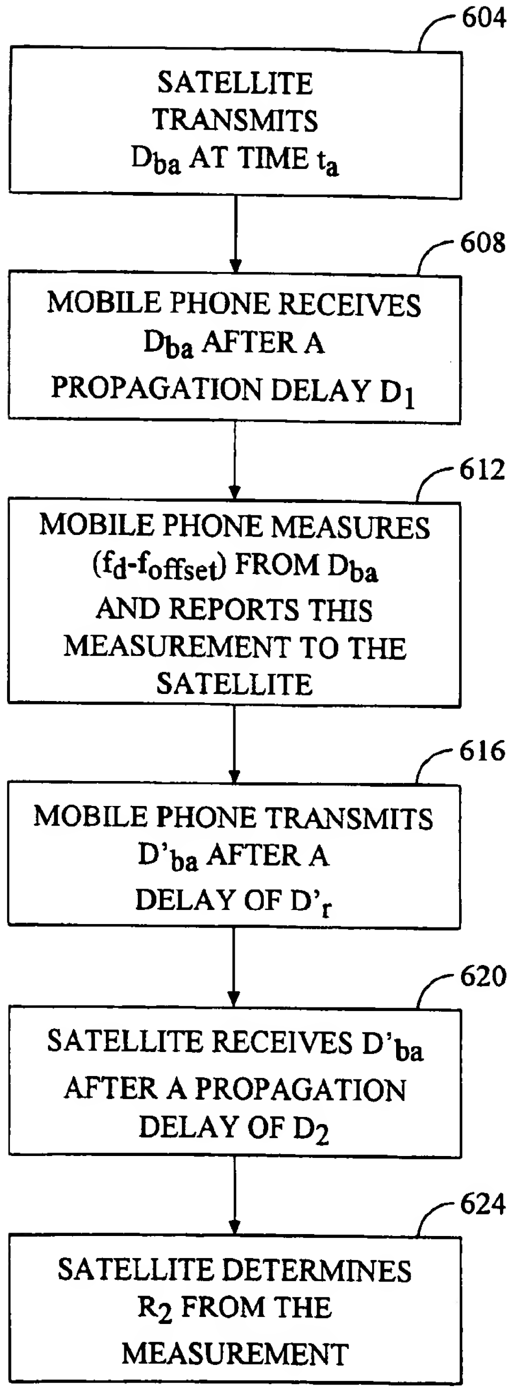

604
SATELLITE TRANSMITS D_{ba} AT TIME t_a
608
MOBILE PHONE RECEIVES D_{ba} AFTER A PROPAGATION DELAY D_1
612
MOBILE PHONE MEASURES (f_d-f_{offset}) FROM D_{ba} AND REPORTS THIS MEASUREMENT TO THE SATELLITE
616
MOBILE PHONE TRANSMITS D'_{ba} AFTER A DELAY OF D'_r
620
SATELLITE RECEIVES D'_{ba} AFTER A PROPAGATION DELAY OF D_2
624
SATELLITE DETERMINES R_2 FROM THE MEASUREMENT

FIG. 5

SYSTEM AND METHOD FOR MEASURING ROUND TRIP DELAY ON THE PAGING AND ACCESS CHANNELS

The Appln claims benefit of Prov. No. 60/127,472 filed Mar. 31, 1999.

BACKGROUND OF THE INVENTION

I. Field of the Invention

The present invention relates generally to wireless communication systems and, more specifically, to a system and method for determining the position of a user terminal that communicates with an earth orbit satellite. Still more specifically, the invention relates to a system and method for measuring the round trip delay on the access channel between a gateway and a user terminal.

II. Description of the Related Art

There is an increasing need in the wireless communications environment for mobile phone location information. For example, with the advent of satellite telephone communications capabilities, it is important to determine the location of a user terminal (the mobile phone) for various reasons including billing and/or geopolitical boundaries. For example, position is needed to select an appropriate ground station or service provider (for example, a telephone company) for providing communication links. A service provider is typically assigned a particular geographic territory, and handles all communication links or calls with users located in that territory. A similar consideration arises when calls must be allocated to service providers based on political boundaries or various contractual relationships.

One industry in particular in which one can see the importance of position information is the commercial trucking industry. In the commercial trucking industry or delivery business, an efficient and accurate method of vehicle position determination is in demand. With ready access to vehicle location information, the trucking company obtains several advantages. The trucking company can keep the customer apprized of location, route and the estimated arrival time of payloads. The trucking company can also use vehicle location information together with empirical data on the effectiveness of routing, thereby determining the most economically efficient routing paths and procedures.

In the past, vehicle location information has been communicated to the trucking company home base by the truck drivers themselves, via telephones, as they reach destinations and stopovers. These location reports are intermittent at best, because they only occur when the truck driver has reached the destination or stopover and can take the time to phone the trucking company home base. These location reports are also quite costly to the trucking company because in effect they cause substantial down time of the freight carrying vehicle. This down time is due to the fact that to make a location report, the tractor driver must remove his vehicle from route, find a telephone which he can use to phone the home base, and take the time to make the location report. This method of location reporting also leaves room for substantial inaccuracies. For example, truck drivers may report incorrect information either mistakenly or intentionally, or report inaccurate estimates of times of arrival and departure.

Presently, the commercial trucking industry is implementing versatile mobile communication terminals for use in their freight hauling tractors. These terminals are capable of providing two-way communication between the trucking company home base and the truck. Typically, the communications are via satellite between the truck and a network communications center or hub.

Using the radio communication capabilities at each mobile terminal to provide vehicle position determination offers great advantages to the commercial trucking industry. Location reports would no longer be intermittent because the trucking company home base could locate a vehicle at will. No down time of the freight hauling vehicle would be required because the communications necessary for determining location could take place while the truck is en route. Also, inaccuracies in location reports would be virtually eliminated because the trucking company home base would be almost instantaneously ascertaining accurate vehicle location information.

However, using the radio communication capabilities at mobile terminals to provide a vehicle or user position is difficult when both the satellite and the vehicle continuously change their position. That is, when low or medium Earth orbiting (LEO or MEO) satellites are used for transferring signals, and when the user or vehicle changes location rapidly or frequently. Due to the orbit of the satellite and the movement of the vehicle, the range between them continuously changes. This makes it difficult to accurately measure the range between the satellite and the mobile phone, and ultimately the location of the phone on the earth's surface. This problem is further discussed below in an example involving two objects that communicate with each other.

Generally, the range between two objects that communicate with each other can be determined in the following way. The first object transmits a first signal and notes the time of transmission. The second object receives the first signal and immediately transmits a second signal. The first object receives the second signal and notes the total time elapsed between the transmission of the first signal and the reception of the second signal. The first object then determines the round trip delay RTD from the relationship RTD=cD/2, where c is the speed of light and D is the total time elapsed between the transmission of the first signal and the reception of the second signal. The range between the two objects can then be determined from RTD.

Unfortunately, this simple relationship (RTD=cD/2) yields an accurate value of R only if (a) the two objects have fixed positions; and (b) the oscillators of both the sending and receiving units are known and stable. In other words, if one of the objects is moving relative to the other object, and/or the oscillator of one of the transmitters is inherently unstable, the simple relationship does not yield an accurate result. Thus, if the first object is a moving object, such as an orbiting communication satellite, and the second object is another moving object, such as a mobile phone mounted on a vehicle, this relationship does not yield an accurate result. Due to the orbit of the satellite and the movement of the mobile phone, the range between the two changes during the time period D. In this scenario, R_1 is the range between the satellite and the mobile phone at the time the satellite transmits the first signal and R_2 is the range at the time the satellite receives the second signal. Needless to say, it is difficult to determine the actual ranges R_1 and R_2 between the mobile phone and the satellite. The ranges can be determined as a function of the round trip delay RTD of signals between a gateway and a mobile phone. A mechanism is therefore needed to accurately determine RTD.

Previously, since it was not possible to accurately determine either R_1 or R_2, which are the ranges from the satellite to the mobile phone at two slightly different time instances, from a measurement that involves their sum, it was difficult

to effectively determine the position of the mobile phone. If a method to effectively determine R_1 or R_2 is provided, it will be possible to determine the position of the mobile phone. Using R_1 (or R_2), and the absolute Doppler, which is equivalent to the range-rate, the position of the mobile phone can be determined. Obtaining the true Doppler, which can be used in determining the range rate, is a subject of U.S. Pat. No. 6,137,441, entitled "Accurate Range and Range Rate Determination in A Satellite Communications System", which is assigned to the assignee of the present invention and is incorporated herein by reference. The technique of that disclosure is only briefly described herein (see Equation 18, infra). Thus an important consequence of determining R_1 and R_2 is that it will then be possible to obtain the position of the mobile phone.

SUMMARY OF THE INVENTION

The present invention is directed to a system and method for determining a round trip delay of signals transmitted between first and second objects, such as a satellite and a mobile telephone, that move relative to each other. In one aspect of the invention, a first signal is transmitted from the first object to the second object. The first signal is received at the second object after a propagation delay D_1, the delay D_1 being the time taken by the first signal to traverse from the first object to the second object. A frequency of the first signal is measured at the second object. The second object then transmits to the first object a second signal containing a report of the measured first frequency. The second signal is received at the first object after a propagation delay D_2, D_2 being the time taken by the second signal to traverse from the second object to the first object. The first object measures a frequency of the second signal. The first object then determines the round trip delay from the measured delays and the measured frequencies of the first and second signals.

In another aspect, the invention is directed to determining a round trip delay of signals transmitted between first and second objects that move relative to each other, in which a first signal is transmitted from the first object. The first signal is received at the second object after a propagation delay D_1. The second object then transmits second signal to the first object, which is received at the first object after a propagation delay D_2. The frequency of the second signal is measured at the first object. The first object then determines the round trip delay from the measured delays and the first signal frequency, the round trip delay being a function of the range traversed by the second signal.

In a still farther aspect, the invention is directed to determining a round trip delay of signals transmitted between first and second objects that move relative to each other, in which a first signal is transmitted from the first object and is received at the second object after a propagation delay D_1. The second object measures a frequency from the first signal and transmits to the first object a second signal containing a report of the measured first signal frequency. The second signal is received at the first object after a propagation delay D_2. The first object then determines the round trip delay from the first signal frequency, the round trip delay being a function of the delay experienced during propagation of the second signal from the second object to the first object.

Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the reference number.

FIG. 1 illustrates a satellite communication system.

FIG. 2 is a time diagram which illustrates the method for determining the distance between a satellite and a mobile phone in accordance with the present invention.

FIG. 3 is a flow diagram illustrating methods 1 and 3 in accordance with the present invention.

FIG. 4 is a flow diagram illustrating method 2 in accordance with the present invention.

FIG. 5 is a flow diagram illustrating method 4 in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. Overview and Discussion of the Invention

In a mobile telephone communications system, and especially in a satellite phone system, it is desirable and important to determine the position of the mobile phone unit (or user terminal). The need for the user terminal position information stems from several considerations. First, the position of the user terminal determines the geographic region in which the user is located, and that determines which service company provides communication service to the user in that region. Thus, determining the location of the user terminal is a pre-requisite to ensuring that the appropriate service provider receives billing credit for providing communication service to the user, that contracted for services or features are provided, or that the most appropriate ground stations are used.

Geopolitical issues based on political boundaries must also be taken into account in some situations. Consider the following scenario where country #1 and country #2 are unfriendly neighbors. A user terminal operates in country #1, where a service provider provides communication service. If a service provider in country #2 incorrectly receives billing credit for the service, it may be difficult to transfer the incorrectly credited amount from country #2 to country #1. If there are various contractual relationships for services for users that travel between the two countries, it may also be difficult to provide the appropriate levels of service the current user location, and so forth.

One conventional approach to position determination is that employed by the U.S. Navy's TRANSIT system. In that system, the user terminal performs continuous Doppler measurements of a signal broadcast by a low-Earth orbit (LEO) satellite. The measurements continue for several minutes. The system usually requires two passes of the satellite, necessitating a wait of more than 100 minutes. In addition, because the position calculations are performed by the user terminal, the satellite must broadcast information regarding its position (also known as "ephemeris"). Although the TRANSIT system is capable of high accuracy (on the order of one meter), the delay required in establishing a position is unacceptable for use in a commercial satellite communications system.

Another conventional approach is that employed by the ARGOS and SARSAT (Search and Rescue Satellite) systems. In that approach, the user terminal transmits an intermittent beacon signal to a receiver on the satellite, which makes frequency measurements of the signal. If the satellite receives more than four beacon signals from the user terminal, it can usually solve for the user terminal's

position. Because the beacon signal is intermittent, an extended Doppler measurement, such as that performed by the TRANSIT system, is unavailable.

Another conventional approach is that employed by the Global Positioning System (GPS). In that approach, each satellite broadcasts a time-stamped signal that includes the satellite's ephemeris. When the user terminal receives a GPS signal, the user terminal measures the transmission delay relative to its own clock and determines the pseudo-range to the transmitting satellite's position. The GPS system requires three satellites for two-dimensional positioning, and a fourth satellite for three-dimensional positioning.

One disadvantage of the GPS approach is that at least three satellites are required for position determination. Another disadvantage of the GPS approach is that, because the calculations are performed by the user terminal, the GPS satellites must broadcast their ephemeris information, and the user terminal must possess the computational resources to perform the required calculations.

A disadvantage of all of the above-described approaches is that the user terminal must have a separate transmitter or receiver, in addition to that required by the communications system, in order to use those approaches.

For these reasons, many in the satellite communications environment have recognized a need for a position determination system capable of rapid position determination. Also, there is a need for a low cost position determination system that accurately determines the position of a user terminal with minimal additional resources at the satellite and the user terminal.

In a satellite communication system, the position of a mobile user terminal can be determined as a function of the range between the satellite and the mobile user terminal and the rate of change of that range. Both are a function of the round trip delay of a signal transmitted from a gateway (fixed ground station transceiver) to a mobile user terminal via a satellite interface and back to the gateway, a frequency measurement made at the mobile user terminal, and a frequency measurement made at the gateway. Because of the movement of a satellite in orbit around the earth and the movement of a mobile phone on the earth's surface, the distance or range between the two changes continuously.

The greatest contribution to the change in geometric relation between the user terminal and the satellite is due to the satellite movement. A LEO satellite travels at an orbital velocity on the order of 16,000 miles per hour or about 4.4 miles per second. The user terminal, by contrast, travels at a typical ground speed of less than 60 miles per hour (about 88 feet per second). The change in position of the user terminal is so small relative to the change in satellite position that the user terminal position change can be effectively ignored. However, due to this relative movement, it is difficult to accurately measure the distance between a mobile phone (or user terminal) and a satellite. Consequently, it is difficult to determine the position of the mobile phone. This creates a problem in wireless communication systems where there is a need to accurately determine the position of the mobile phone.

The present invention provides a solution to this problem. The present invention provides a method for determining the round trip delay of a signal transmitted between a gateway and the mobile phone. From this information, the distance or range between the mobile phone and the satellite can be determined. An important consequence of the present invention is that it also provides a method for subsequently accurately determining the position of the mobile phone.

It must be noted that the utility of the present invention is not limited to satellites and mobile phones. In fact, in a broad sense, the present invention can be utilized to determine the range between any two moving objects that communicate with each other.

2. Example Environment

Before describing the invention in detail, it is useful to describe an example environment in which the invention can be implemented. In a broad sense, the invention can be implemented in a variety of communication systems. One such communication system **100** is illustrated in FIG. 1. Specifically, FIG. 1 shows a satellite **104** moving in an orbital path O. A mobile station, phone, or user terminal **108** is located effectively at or near the surface of the earth E. Mobile phone **108** comprises a wireless communication device such as, but not limited to, a cellular telephone, or a data or position determination transceiver, and can be hand-held or vehicle-mounted as desired. However, it is also understood that the teachings of the invention may be applicable to fixed units where position determination is desired. The satellite transmits signals to, and receives signals from, mobile phone **108**, called a user terminal (UT), through a fixed ground station **112**, called a gateway (GW).

FIG. **1** is presented to illustrate the difficulties associated with measuring the distance between satellite **104** and mobile phone **108** in a dynamic environment. This dynamic environment is created by the movement of satellite **104** in its orbit O and the movement of UT **108** on the earth's surface, where the distance between satellite **104** and UT **108** continuously changes. This makes it difficult to accurately measure the distance between them and ultimately the position of UT **108**.

In this example environment and hereinafter, gateway **112** communicates with UT **108** via satellite **104**. At a time t_1, gateway **112** transmits a forward uplink paging signal S_{fu} to satellite **104**. The reason for designating this time as t_{-1} will be made clear below. At time t_0, circuitry in satellite **104** converts signal S_{fu} to a forward downlink signal S_{fd} and transmits it to UT **108**. UT **108** receives signal S_{fd} at time t_1. UT **108** immediately transmits a reverse uplink signal S_{ru} which is received by satellite **104** at time t_2. Satellite **104** converts signal S_{ru} to a reverse downlink signal S_{rd} which is then transmitted to gateway **112**, where it is received by gateway **112** at time t_3. Signals S_{fu} and S_{fd} are transmitted on one or more paging channels. Signals S_{ru} and S_{rd} are transmitted on an access channel which is used by user terminals to "access" a gateway. A user terminal accesses a gateway to register with the system, to place a call, or to acknowledge a paging request sent by the gateway, by transmitting data on the access channel that contains an access message.

During the elapsed time period between times t_0 and t_2, that is, t_2-t_0, satellite **104** has changed its position. UT **108** may also have changed its position. However, for reasons mentioned above, the position change of UT **108** may be ignored As a result, the range between UT 108 and satellite **104** changes from R_1 to R_2. As noted above, the largest contribution to this change in position is due to the movement of the satellite in orbit.

The present invention is described in terms of this example environment. Description in these terms is provided for convenience only. It is not intended that the invention be limited to applications in this example environment. In fact, after reading the following description, it will become apparent to a person skilled in the relevant art how to implement the invention in alternative environments.

3. The Present Invention

In the present invention, gateway 112 transmits paging signal S_{fu} at periodic intervals. For purposes of this invention, signal S_{fu} is deemed to be transmitted by gateway 112 at time t_{-1} and received by satellite 104 at time t_0. A transmitter, located at gateway 112, pre-corrects the frequency of forward link signal S_{fu} to compensate for the Doppler shift due to the relative motion between satellite 104 and gateway 112. Because the relative motion of satellite 104 with respect to gateway 112 is well known, the signal is compensated so that when the signal reaches satellite 104, the signal does not appear to have experienced any Doppler shift due to the relative motion. In other words, signal S_{fu} is pre-corrected by the transmitter to compensate for the Doppler shift.

In addition, the transmitter at gateway 112 pre-corrects the timing of signal S_{fu}. Signal delays on the order of 5–15 ms can occur between the time of transmission from gateway 112 to the time of receipt by satellite 104 and vice versa. Similar delays occur in signals transmitted between satellite 104 and UT 108. In the forward link, only the timing of signal S_{fu} is pre-corrected. The timing of signal S_{fu} is continuously adjusted so that the signal arrives at the satellite with known timing or at a known time, referred to as satellite time. Thus, the gateway transmitter adjusts the timing of signal S_{fu} transmitted to satellite 104 so that the signal is synchronized at the satellite at a predetermined time regardless of the distance between the gateway and the satellite.

One result of pre-correcting the timing is that the timing uncertainty at the user terminal due to variation in propagation delay is reduced. Because the timing of the uplink portion of the forward link signal is known, the only uncertainty due to propagation delay occurs in the downlink portion of the forward link. Thus, by pre-correcting the timing, the timing uncertainty in the forward link signal is reduced by approximately one half.

Analogous post-correction adjustments to frequency and timing are made to signal S_{rd}, transmitted from satellite 104 to gateway 112. However, since the range between satellite 104 and UT 108 is not known, the system cannot pre-correct signals S_{fd} or S_{ru}.

Continuously pre-correcting and post-correcting the timing of the signal in a CDMA systems results in each code in a PN spreading code sequence arriving at any particular satellite or gateway at the same time as other satellites or gateways regardless of the distance between the gateway and the satellite. In other words, the uplink portion of the forward link signal at the satellite and the downlink portion of the reverse link signal at the gateway do not exhibit any code Doppler. Preferably, therefore, signal S_{fu} is pre-corrected for Doppler and time so that the signal that is received at satellite 104 is effectively seen by satellite 104 as though it had been transmitted instantaneously from gateway 112. Thus the time at which satellite 104 receives signal S_{fu} is designated as time t_0.

Pre-correction and post-correction of signals is described in detail in U.S. patent application Ser. No. 08/723,490, entitled "Time and Frequency Correction For Non-Geostationary Satellite Communications System," filed Sep. 30, 1996, in the name of S. Kremm, the disclosure of which is incorporated by reference herein in its entirety.

As noted above, gateway 112 transmits a message as a carrier modulated signal S_{fu} on the paging channel of the satellite communications system at time t_{-1}. This message contains at least the time of transmission of the message adjusted for the pre-correction performed by gateway 112. Satellite 104 sends carrier modulated signal S_{fd} at time t_0. Signal S_{fd} is received by UT 108 at time t_1. Upon receipt of signal S_{fd}, UT 108 immediately transmits a carrier modulated reply message as signal S_{ru}. The reply message contains information to be used by gateway 112 to determine round trip delay. An important piece of information contained in the reply message is the time of receipt of signal S_{fd} as perceived by UT 108. However, this time information cannot be used directly by gateway 112. The time of receipt of signal S_{fd} at UT 108 is measured according to a local user terminal time based on signals provided by a local oscillator in UT 108. The problem is that the local oscillator in UT 108 is inherently unstable or variable. That is, to reduce costs and simplify circuits, UT oscillators tend to be relative inexpensive and subject to drift and other variations which make them inaccurate, creating errors in the output signal, and, thus, any time measurements based on such signals. The crystal oscillator in UT 108 may have a frequency error on the order of 10 ppm. This can introduce significant errors into the time measurements.

The reply message transmitted by UT 108 on signal S_{ru} is received at satellite 104 at time t_2 and at gateway 112 as signal S_{rd} at time t_3. The reply message contains information about the time at which UT 108 received the transmitted message sent by gateway 112, as that time is perceived by UT 108. Gateway 112 waits for a period of time after it begins to receive signal S_{rd} before taking measurements of the information contained in signal S_{rd} at time t_4. The goal is to wait as long as possible after the start of the message. This ensures a more accurate measurement because the time tracking circuitry locks in more solidly the longer the wait.

There is a problem with waiting too long to begin measuring the timing of the message in signal S_{rd}. UT 108 can start its transmission at the point at which UT 108 perceives time. However, UT 108 must construct its message and put into the message the time at which it received the message in signal S_{fd} and the time at which it begins transmission of the reply message on signal S_{rd}. What goes into the reply message is the time T_{RX} at which UT 108 received the message on signal S_{fd}. This is equal to the time T_{TX} at which UT 108 transmits its message on signal S_{ru}.

The reply message signal generated by UT 108 has certain characteristics. One is that data is clocked out at a certain rate. Another characteristic is that the information signal is modulated onto a carrier frequency. Both characteristics are derived from the same inherently error prone crystal oscillator in UT 108. If gateway 112 can measure the frequency of the signal generated by UT 108 and can determine what that frequency should be, and by measurement gateway 112 knows what that frequency actually is, then gateway 112 can determine the characteristics of the local oscillator of UT 108 based on the difference in frequency measurement. Now, knowing the information about UT 108's local oscillator, which is also being used to clock the data transmitted on signals S_{ru} and S_{rd}, gateway 112 can use that information to determine the true length of the message transmitted by UT 108. From that information, gateway 112 can then determine a better estimate of the actual beginning of the message transmitted by UT 108.

A feature of the present invention is the recognition that gateway 112 can use the frequency measurement it makes on the signal received from UT 108, and compare that measured frequency to a theoretical carrier frequency to determine a best estimate of the actual time at which UT 108 began transmitting its message. By itself, however, this measurement does not provide the necessary information to accurately determine the position of UT 108. Merely measuring the frequency of the signal transmitted by UT 108

does not take into account the Doppler effect caused by the movement of satellite **104** in its orbit. The Doppler effect causes the frequency of the received signal to differ from the frequency of the transmitted signal. Moreover, each of signals S_{fd}, S_{ru}, and S_{rd} is affected by Doppler. The Doppler itself changes. This creates a second order effect which can either be accounted for in the algorithm or method steps of the invention, or can be ignored.

The method and system of the present invention makes one set of measurements needed for position determination. The present invention compensates for the errors in the internal clock timing of UT **108** to determine the distance between UT **108** and satellite **104** at some instant in time. From that, the position of UT **108** can then be determined. Gateway **112** begins to measure the signal received from UT **108** at time t_4. Gateway **112** then measures the length of the message received from UT **108**. Gateway **112** then subtracts the message length from the time of measurement to determine the start time of the received message. This measurement provides an accurate start/receive time t_3 of the received signal. From this information, the distance between satellite **104** and UT **108** can be calculated.

Another measurement that is made is the carrier frequency of the signal received at gateway **112**. By correcting for Doppler, the transmitting frequency of UT **108** can then be determined.

In summary, the following information is either known or can be calculated or estimated:

1. The frequency of signal S_{rd} received at gateway **112** can be measured.
2. The distance between gateway **112** and satellite **104** is known to a good approximation.
3. The frequency of signal S_{fu} from gateway **112** to satellite **104** is known.
4. Doppler between gateway **112** and satellite **104** is known.
5. The frequency of signal S_{ru} at satellite **104** can be calculated from the above known or measured information.
6. Doppler of signal S_{fd} from satellite **104** to UT **108** can be reasonably estimated.

From the foregoing known, measurable or reasonably estimable information, it is possible to correct back to estimate the transmitting frequency of UT **108**.

Satellite **104** contains a translator for translating forward uplink frequencies to forward downlink frequencies and for translating reverse uplink frequencies to reverse downlink frequencies. While these frequencies are assumed to be different in the present example system, they can be the same in some communication system designs to simplify the transfer process, which will still suffer from Doppler. In the satellite the translator translates not only the nominal frequencies; it also translates the Doppler frequencies. Thus, the actual frequency that is translated is the nominal plus Doppler (f_n+f_d).

The present invention is primarily intended to operate in an environment in which the satellite acts only as a "bent pipe" with a frequency translator. The satellites contemplated for use with this invention do not have intelligence built into them to accomplish such tasks as correcting for Doppler. In addition, it would be very difficult for the satellite to correct for Doppler on the reverse uplink and forward downlink frequencies. This is because the satellite does not know the location of the user terminal. Furthermore, the satellite is receiving signals from and sending signals to multiple user terminals simultaneously. Even if the locations of the user terminals were known, the satellite could not correct for Doppler for each concurrent transmission to and from multiple user terminals. It would be apparent to one skilled in the relevant arts to make the appropriate modifications in the relevant algorithms should it be desired to utilize the techniques of the present invention in a satellite system that has the intelligence to correct for Doppler on one or both of the uplink and downlink signals.

FIG. **2** shows a timing diagram of the forward downlink and reverse uplink signals. In the figure, point t_a represents the timing point of the forward downlink signal which is selected by UT **108** as the marker at which UT **108** begins its transmission of a reply signal to be used by gateway **112** to determine round trip signal delay. Time marker t_a is generated by gateway **112** and its temporal position within signals S_{fu} and S_{fd} is known by the gateway. Marker t_a is transmitted from satellite **104** as part of signal S_{fd} and is received at UT **108** after a delay D_1. Delay D_1 is a function of the distance between satellite **104** and UT **108**. Immediately upon receipt by UT **108** of marker t_a, and subject to a small built in delay D_r on the order of not more than about 200 microseconds to avoid collisions between signals from different UTs, UT **108** begins transmitting a reply signal on its reverse uplink channel.

While various access signal formats can be used, a preferred structure for the access signal transmitted by UT **108** contains three parts: a preliminary preamble (block A), a main preamble (block B), and a data block (block C). Therefore, each access message is divided into a preamble address portion (blocks A and B in FIG. **2**) and a data portion (data block C in FIG. **2**). The transmission of each message preamble portion precedes the transmission of the data portion by a predetermined period of time to allow a gateway to adjust its tracking circuitry and synchronize with the received signal before the arrival of the data portion. The use of this type of signal is discussed in more detail in U.S. patent application Ser. No. 09/098,631, entitled "Rapid Signal Acquisition And Synchronization For Access Transmissions," filed Jun. 16, 1998 the disclosure of which is incorporated by reference herein in its entirety.

Signal S_{ru} is received at satellite **104** time t_m. This is the point at which the end of the preamble (blocks A and B) is received by satellite **104**. The delay period D_2 equals the satellite path delay at t_m, that is the delay in transmission time due to the distance between UT **108** and satellite **104**. The signal comprising preamble blocks A and B and data block C is actually received at gateway **112**. However, since gateway **112** knows at any one time the exact location of satellite **104**, the gateway circuitry can perform post-correction on reverse downlink signal S_{rd} to determine when the end of preamble blocks A and B of signal S_{ru} was received at satellite **104**. Data block C contains information about time marker t_a. Gateway **112** also knows the length of blocks A and B. Gateway **112** does not know the length of gaps D_1 or D_2 or the frequency of the signal transmitted by UT **108**.

In addition, gateway **112** must take into account the Doppler frequency. An assumption can be made that the Doppler does not change significantly over the course of the transmission from UT **108** to satellite **104**.

According to the present invention, gateway **112** transmits a signal on the paging channel. The signal is received by UT **108** after a propagation delay D_1. The distance traversed by the signal between gateway **112** and UT **108** is referred to as the forward distance.

The UT transmits a preamble portion of an access message after a predetermined guard interval D'_r in order to

avoid access signal collisions. The preamble portion is received by gateway **112** after a propagation delay D_2. The distance traversed by the preamble portion between UT **108** and satellite **104** is referred to as the reverse distance.

When UT **108** receives the signal transmitted by the gateway, it measures a first frequency from the transmitted gateway signal and relays the measurement back to gateway **112** in the data portion of the access message. Alternatively, to reduce complexity and simplify the process, the UT measures a relative offset of the received signal from the internal oscillator frequency and provides this measured offset as data to the gateway. Gateway **112** measures a second frequency from the data portion.

One goal is to solve for D_{meas}, which is the measured delay between the time t_b that the end of the preamble portion is expected to be received and the time t_m when the end of the preamble portion is actually received at satellite **104**.

The present invention can be categorized into four separate but related methods. Method 1 is a high accuracy implementation requiring frequency measurements at both the UT and the gateway and yields the satellite to UT range at the time instance corresponding to the start of the access slot at the gateway. Method 2 is a less accurate approximation of method 1. Method 2 requires a frequency measurement only at the gateway. Method 3 is a high accuracy implementation requiring frequency measurements at both the UT and the gateway and yields the satellite to UT range at the time instance at which the end of the preamble portion (t_m) is received at the gateway. Method 4 is a less accurate version of method 3. Method 4 requires a frequency measurement only at the UT. These methods are discussed in detail below, with reference to the following index:

t_a=start of access probe on paging channel at satellite **104** (and at gateway **112**, as a result of pre-correction).

t_b=t_a+D_{ba}

t_m=measurement at satellite **104** (and at gateway **112**, as a result of post correction).

D_{ba}=predetermined duration of the preamble of signal S_{ru} (for example, the nominal length of preamble blocks A and B of the access probe).

D'_{ba}=the predetermined duration of the preamble of signal S_{ru} as generated with error by the UT clock error.

D_r=random delay between receipt of signal S_{fd} at UT **108** and the start of transmission of signal S_{ru} (for collision avoidance).

D'_r=random delay as actually generated by UT **108** due to clock error.

D_1=satellite-UT path delay at t_a

D_2=UT-satellite path delay at t_m (satellite-gateway delay is known and pre-corrected.

f_F=nominal carrier frequency transmitted from the satellite to the UT (known as the "forward frequency")

f_R=nominal carrier frequency transmitted from the UT to the satellite (known as the "reverse frequency")

f_{offset}=frequency offset of the UT at the frequency f_F

f_D=Doppler shift of frequency f_F

f_D/f_0=normalized Doppler frequency

D_{meas}=the measured delay between the expected end of the preamble portion and the actual received end of the preamble portion.

R_1=$R_{sat\text{-}UT}(t_a)$, (t_a) is a well defined time instance at gateway **112**.

R_2=$R_{sat\text{-}UT}(t_m)$, t_m, when gateway **112** identifies the end of the received the preamble portion, is also a well defined time instance at gateway **112**.

$\dot{R}_1$=$\dot{R}_2$=$\dot{R}$, (the range rate is assumed constant) C=velocity of light

METHOD 1

Referring now to FIG. **2**, at time t_a, satellite **104** transmits a signal S_{fd} on the paging channel. UT **108** receives signal S_{fd} after a delay period D_1. D_1, which is also known as the satellite-phone path delay, is given by the relationship

$$D_1 = \frac{R_1}{c}$$

where

R_1=the distance or range between satellite **104** and UT **108** at the time of transmission of signal S_{fd}.

UT **108** measures the frequency of the received signal S_{fd}. This measurement is offset from the actual transmitted frequency due to two factors: local oscillator error of the UT's internal clock; and the Doppler effect caused by the relative movement between satellite **104** and UT **108**. The measured value of the frequency of signal S_{fd} as perceived by UT **108** is encoded into the data portion of signal S_{ru} and transmitted back to gateway **112**. Signal S_{ru} is transmitted at the exact moment that time marker t_a is received at the antenna of UT **108**, plus, as noted above, a short delay D_r to avoid signal collision.

The signal S_{ru}, comprising the preamble portion and data portion of the access message, is received at gateway **112** at time t_m. From the information contained in signal S_{ru}, gateway **112** can determine the time at which the end of the preamble portion was expected to arrive at gateway **112** (time t_b) and from this, gateway **112** can determine the round trip delay (RTD).

Note that during the time period between t_a and t_b, due to the movement of satellite **104** in orbit, and also the movement of UT **108**, both satellite **104** and UT **108** change their position. Consequently, the range between satellite **104** and UT **108** changes from R_1 to R_2.

At time t_m, satellite **104** receives the end of the preamble portion. The UT-satellite path delay is D_2 which is given by the relationship

$$D_2 = \frac{R_2}{c}.$$

Also at time t_m, satellite **104** measures (f_D+f_{offset}) from signal S_{ru}.

As noted previously, the present invention provides a method to determine the round trip delay of the access channel signal in a dynamic environment created by the movement of satellite **104** and UT **108**. This movement of satellite **104** and UT **108** introduces Doppler frequency f_D in signal S_{fd} and signal S_{ru}. The Doppler frequency f_D is given by the relationship:

$$f_D = \dot{R}\frac{f_F}{c} \quad \text{or,} \quad \dot{R} = -f_D\frac{c}{f_F}$$

where $\dot{R}$=dR/dt, f_F is the oscillator frequency at satellite **104** and c is the speed of light.

In order to simplify the analysis, an assumption is made that the rate of change of R_1 and R_2 is constant, i.e., $\dot{R}_1$=$\dot{R}_2$=$\dot{R}$. This implies that the Doppler frequency f_D is also constant. Now, from FIG. **2** we get:

$$R_2 = R_1 + \dot{R}(t_m - t_a) \tag{1}$$

Next, we substitute $(t_m-t_a)=(D'_{ba}+D'_r+D_1+D_2)$ in equation (1) and get:

$$R_2 = R_1 + \dot{R}(D'_{ba} + D'_r + D_1 + D_2) \tag{2}$$

Next, we substitute $-cf_D/f_F$ in equation (2) and get

$$R_2 = R_1 - c\frac{f_D}{f_F}(D'_{ba} + D'_r + D_1 + D_2) \tag{3}$$

$$D_{meas}=D_1+D'_{ba}+D'_r+D_2-D_{ba} \tag{4}$$

Next, we substitute $D'_{ba}=D_{ba}(1-f_{offset}/f_F)$ and $D'_r=D_r(1-f_{offset}/f_F)$ in equation (4) and get

$$D_{meas} = \frac{R_1}{c} + (D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) + \frac{R_1}{c} - \frac{f_D}{f_F}\left[(D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) + \frac{R_1}{c} + \frac{R_2}{c}\right] - D_{ba} \tag{5}$$

or,

$$D_{meas} = \frac{R_1}{c} + (D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) - D_{ba} + \frac{R_1}{c} - \frac{f_D}{f_F}\left[(D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) + \frac{R_1}{c} + \frac{R_1}{c} - \frac{f_D}{f_F}\left[(D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) + \frac{R_1}{c} + \frac{R_2}{c}\right]\right] \tag{6}$$

or,

$$D_{meas} = 2\frac{R_1}{c}\left(1 - \frac{f_D}{f_F}\right) + (D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right)\left(1 - \frac{f_D}{f_F}\right) - D_{ba} + \left(\frac{f_D}{f_F}\right)^2\left[(D_{ba} + D_r)\left(1 - \frac{f_{offset}}{f_F}\right) + \frac{R_1}{c} + \frac{R_2}{c}\right] \tag{7}$$

In order to further simplify equation (7), we make the following approximation. Since both the local oscillator error frequency f_{offset} and the Doppler frequency f_D are very small, we ignore the terms multiplied by f_{offset}/f_0^2 or by f_D^2/f_0. As a result, we get

$$D_{meas} = 2\frac{R_1}{c}\left(1 - \frac{f_D}{f_F}\right) + (D_{ba} + D_r)\left(1 - \frac{f_D + f_{offset}}{f_F}\right) - D_{ba} \tag{8}$$

Next, we rearrange equation (8) and get

$$R_1 = \frac{c}{2\left(1 - \frac{f_D}{f_F}\right)}\left[D_{meas} - D_r + (D_{ba} + D_r)\frac{f_D + f_{offset}}{f_F}\right] \tag{9}$$

Note that in Equation (9) which represents method 1, R_1 is given as a function of f_D, f_{offset}, f_F, D_{meas}, D_r and D_{ba}. Except for f_D and f_{offset}, which are measured at gateway **112** and UT **108**, the other terms are known. In particular, method 1 requires (f_D+f_{offset}) and f_D. Since, (f_D+f_{offset}) is measured at gateway **112**, it is readily available at gateway **112**. However, f_D by itself is not readily available at gateway **112**. In order to get f_D, UT **108** must measure (f_D-f_{offset}) and report the measurement to gateway **112**. Then, from (f_D+f_{offset}) and (f_D-f_{offset}), f_D is determined.

The principal advantage of method 1 is that it provides an accurate result for R_1. However, since method 1 requires both f_D and (f_D+f_{offset}), it requires measurements at both gateway **112** and UT **108**.

The robustness of Equation (9) can be easily verified by testing whether it provides a correct formula when two objects have fixed positions. When satellite **104** and UT **108** have fixed positions, f_D and f_{offset} are both zero. In that scenario, Equation (9) is reduced to the following equation:

$$R_1 = c\frac{D_{meas} - D_r}{2} \tag{10}$$

Thus, Equation (10) provides a correct formula when two objects have fixed positions. In reality, however, Equation (10) is solely used for verifying the various assumptions made in arriving at Equation (9) since satellite **104** and UT **108** do not have fixed positions.

METHOD 2

Since the maximum normalized Doppler shift (approximately 20 ppm) is considerably larger than normalized offset frequency, we can approximate f_D/f_F by

$$\frac{(f_D + f_{offset})}{f_F}$$

in method 1 to get:

$$R_1 \approx \frac{c}{2\left(1 - \frac{f_D + f_{offset}}{f_F}\right)}\left[D_{meas} - D_r + (D_{ba} + D_r)\frac{f_D + f_{offset}}{f_F}\right] \tag{11}$$

Note that method 2, which is given by Equation (11), requires only one measurement at gateway **112**. Thus, method 2 eliminates an additional measurement at UT **108**. However, method 2 is less accurate than method 1, and the worst case normalized error in method 2 is f_{offset}/f_F.

METHOD 3

As noted previously, method 3 provides a solution for R_2. Method 3 differs from method 1 by expressing R_1 as a function of R_2. From equation (1), we get:

$$R_1 = R_2 - \dot{R}(t_m - t_a) = R_2 - \dot{R}(D'_{ba} + D_r + D_1 + D_2) \tag{12}$$

Following an analysis similar to the one performed in method 1, we get:

$$R_1 \approx \frac{c}{2\left(1 + \frac{f_D}{f_F}\right)}\left[D_{meas} - D_r - (D_{ba} + D_r)f_D - \frac{f_{offset}}{f_F}\right] \tag{13}$$

Note that Equation (13) requires both (f_D-f_{offset}) and f_D. Since (f_D-f_{offset}) is measured at the UT, it is readily available at the UT. However, f_D is available only if in addition to the measurement at the UT, the gateway performs its own measurement. The principal advantage of method 3 is that it provides an accurate result for R_2. However, method 3 requires measurements at both the gateway and the UT.

The robustness of Method 3 can also be verified by assuming that the satellite **104** and UT **108** have fixed positions. When satellite **104** and UT **108** are assumed to have fixed positions, we get:

$$R_2 = c\frac{D_{meas} - D_r}{2} \tag{14}$$

METHOD 4

Since the maximum normalized Doppler shift (approximately 20 ppm) is much greater than normalized offset frequency, we can approximate f_D/f_F by in

$$\frac{(f_D - f_{offset})}{f_F}$$

method 3 to get:

$$R_2 \approx \frac{c}{2\left(1 + \frac{f_D + f_{offset}}{f_F}\right)}\left[D_{meas} - D_r + (D_{ba} + D_r)\frac{f_D + f_{offset}}{f_F}\right] \tag{15}$$

Note that method 4 requires only one measurement at the UT. Thus, method 4 eliminates an additional measurement at the gateway.

Two measured frequencies are available at gateway **112**, as follows:

1) the reported measurement from the UT:

$$f_{meas,UT} = f_F\left(-\frac{\dot{R}}{c} - \frac{f_{offset}}{f_F}\right) \tag{16}$$

where f_F=the forward link carrier frequency (2500 MHz).

2) the measurement performed at the GW itself:

$$f_{meas,GW} = f_R\left(-\frac{\dot{R}}{c} + \frac{f_{offset}}{f_F}\right) \tag{17}$$

where f_R=the reverse link carrier frequency (1600 MHz).

Adding and subtracting (16) and (17) yields both the UT offset and the range-rate:

$$\dot{R} = -\frac{c}{2}\left(\frac{f_{meas,GW}}{f_R} + \frac{f_{meas,UT}}{f_F}\right) \tag{18}$$

$$\frac{f_{offset}}{f_F} = \frac{1}{2}\left(\frac{f_{meas,GW}}{f_R} - \frac{f_{meas,UT}}{f_f}\right) \tag{19}$$

The Doppler and offset frequencies at the forward frequency are:

$$f_D \ @ \ f_F = -\frac{\dot{R}}{c}f_F = \frac{1}{2}\left(\frac{f_F}{f_R}f_{meas,GW} + f_{meas,UT}\right) \tag{20}$$

$$f_{offset} \ @ \ f_F = \frac{1}{2}\left(\frac{f_F}{f_R}f_{meas,GW} - f_{meas,UT}\right) \tag{21}$$

The difference between the time of the GW and the UT measurements is accounted for as follows:

$$\dot{R}_1 \neq \dot{R}_2$$

$$f_{meas,UT} = f_F\left(-\frac{\dot{R}_1}{c} - \frac{f_{offset}}{f_F}\right) \tag{22}$$

$$f_{meas,GW} = f_R\left(-\frac{\dot{R}_2}{c} + \frac{f_{offset}}{f_F}\right) \tag{23}$$

$$\bar{f}_D \ @ \ f_F = -\frac{\dot{R}_1 + \dot{R}_2}{2c}f_F = \frac{1}{2}\left(\frac{f_F}{f_R}f_{meas,GW} + f_{meas,UT}\right) \tag{24}$$

$$f_{offset} \ @ \ f_F = \frac{1}{2}\left(\frac{f_F}{f_R}f_{meas,GW} - f_{meas,UT}\right) + \frac{f_F}{2c}(\dot{R}_2 - \dot{R}_1) \approx \frac{1}{2}\left(\frac{f_F}{f_R}f_{meas,GW} - f_{meas,UT}\right) \tag{25}$$

FIG. **3** is a flow diagram illustrating both methods 1 and 3. Referring now to FIG. **3**, in step **404**, gateway **112** transmits signal S_{fd} at time t_a. Next, in step **408**, UT **108** receives signal S_{fd} after a propagation delay D_1. Next, in step **412**, UT **108** measures (f_D-f_{offset}) from signal S_{fd} and reports the measurement to the gateway. In step **416**, UT **108** transmits signal S_{ru} after a delay of D'_r. In step **420**, gateway **112** receives signal S_{ru} after a propagation delay of D_2. In step **424**, gateway **112** measures (f_D+f_{offset}) from signal S_{ru}. Finally, in step **428**, gateway **112** determines R_1 and R_2 from the measurements.

FIG. **4** is a flow diagram illustrating the steps involved in method 2 and FIG. **5** is a flow diagram illustrating the steps involved in method 4. Since the steps involved in methods 2 and 4 are very similar to the steps in methods 1 and 3, they will not be separately described herein.

The present invention contemplates that the ranges R_1 and R_2 and the location of the UT **108** can also be determined at the UT. In one alternate embodiment of the present invention, UT **108** measures (f_D-f_{offset}) from D_{ba}. Gateway **112** measures (f_D-f_{offset}) from D'_{ba} and reports the measurement to UT **108**. Using these two measurements, UT **108** determines R_1 and R_2. Finally, UT **108** determines its own location using the method described before.

In summary, the present invention provides four different methods for determining R_1 or R_2. Methods 1 and 2 provide solutions for R_1. Method 1 provides a highly accurate solution for R_1, but it requires two frequency measurements. Method 2 requires only one frequency measurement, but is less accurate than method 1. Methods 3 and 4 provide solutions for R_2. Method 3 provides a highly accurate solution for R_2, but it also requires two frequency measurements. Method 4 requires only one frequency measurement, but it is less accurate than method 3.

The present invention also provides a method to determine the range-rate between the satellite and the UT (Equation 8). The information of the RTD and the range-rate is sufficient to determine the location of a UT on the earth's surface.

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What we claim as our invention is:

1. A method for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

transmitting a first signal from said first object to said second object;

receiving said first signal at said second object after a propagation delay D_1, said delay D_1 being the time taken by said first signal to traverse from said first object to said second object;

measuring at said second object, a first frequency associated with said first signal;

transmitting from said second object to said first object, a second signal containing a report of the measured first frequency;

receiving said second signal at said first object after a propagation delay D_2, D_2 being the time taken by said second signal to traverse from said second object to said first object;

measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal;

measuring at said first object, a second frequency associated with said second signal; and

determining said round trip delay at said first object, said round trip delay being based upon (i) the measured first and second frequencies and (ii) the measured elapsed time.

2. The method according to claim **1**, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

3. The method according to claim **1**, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

4. The method according to claim **1**, wherein said first frequency is $(f_D - f_{offset})$, where f_D is the Doppler frequency of the first signal transmitted between said first object and said second object and f_{offset} is the second object's local oscillator error frequency.

5. The method according to claim **1**, wherein said second frequency is $(f_D + f_{offset})$, where f_D is the Doppler frequency the second signal transmitted between said second object and said first object and f_{offset} is the second object's local oscillator error frequency.

6. The method of claim **1**, wherein the transmitting at said second object occurs within a predetermined time of receipt of said first signal.

7. A method for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising the steps of:

transmitting a first signal from said first object;

receiving said first signal at said second object after a propagation delay D_1;

transmitting a second signal from said second object to said first object upon receipt of said first signal;

receiving said second signal at said first object after a propagation delay D_2;

measuring, at said first object, a frequency from said second signal; and

determining, at said first object, said round trip delay from said frequency, said round trip delay being a function of the delay experienced by said first signal during propagation from said first object to said second object.

8. The method according to claim **7**, wherein said measured frequency is $(f_D + f_{offset})$, where f_D is the Doppler frequency of the second signal and f_{offset} is the second object's local oscillator error frequency.

9. The method according to claim **7**, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

10. The method according to claim **7**, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

11. A method for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising the steps of:

transmitting a first signal from said first object;

receiving said first signal at said second object after a propagation delay D_1;

measuring, at said second object, a frequency from said first signal;

transmitting, from said second object to said first object, a second signal containing a report of the measured first signal frequency;

receiving, at said first object, said second signal after a propagation delay D_2; and

determining, at said first object, said round trip delay from said first signal frequency, said round trip delay being a function of the delay experienced during propagation of said second signal from said second object to said first object.

12. The method according to claim **11**, wherein said frequency is $(f_D - f_{offset})$, where f_D is the Doppler frequency of the first signal and f_{offset} is the second object's local oscillator error frequency.

13. The method according to claim **11**, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

14. The method according to claim **11**, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

15. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay D_1, said delay D_1 being the time taken by said first signal to traverse from said first object to said second object;

means for measuring at said second object, a first frequency associated with said first signal;

means for transmitting from said second object to said first object, a second signal containing a report of the measured first frequency;

means for receiving said second signal at said first object after a propagation delay D_2, D_2 being the time taken by said second signal to traverse from said second object to said first object;

means for measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal;

means for measuring at said first object, a second frequency from said second signal; and

means for determining said round trip delay at said first object, said round trip delay being based upon (i) the measured first and second frequencies and (ii) the measured elapsed time.

16. The system according to claim **15**, wherein said first frequency is $(f_D - f_{offset})$, where f_D is the Doppler frequency of the first signal and f_{offset} is the second object's local oscillator error frequency.

17. The system according to claim **16**, wherein said second frequency is $(f_D + f_{offset})$, where f_D is the Doppler

frequency and f_{offset} is the second object's local oscillator error frequency.

18. The system according to claim 15, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

19. The system according to claim 15, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

20. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay

means for measuring at said second object, a first frequency associated with said first signal;

means for transmitting from said second object to said first object, a second signal containing a report of the measured first frequency;

means for receiving said second signal at said first object after a propagation delay of D_2;

means for measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal;

means for measuring at said first object, a second frequency associated with said second signal; and

means for determining said round trip delay at said first object, said round trip delay being based upon (i) the measured first and second frequencies and (ii) the measured elapsed time.

21. The system according to claim 20, wherein said first frequency is (f_D-f_{offset}), where f_D is the Doppler frequency of said first signal and f_{offset} is the second object's local oscillator error frequency.

22. The system according to claim 21, wherein said second frequency is (f_D+f_{offset}), where f_D is the Doppler frequency of said second signal and f_{offset} is the second object's local oscillator error frequency.

23. The system according to claim 20, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

24. The system according to claim 20, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

25. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay D_1;

means for transmitting a second signal from said second object to said first object upon receipt by said second object of said first signal;

means for receiving said second signal at said first object after a propagation delay D_2;

means for measuring, at said first object, a frequency from said second signal; and

means for determining, at said first object, said round trip delay from said frequency, said round trip delay being a function of the delay experienced during propagation of said first signal from said first object to said second object.

26. The system according to claim 25, wherein said frequency is (f_D+f_{offset}), where f_D is the Doppler frequency of said second signal and f_{offset} is the second object's local oscillator error frequency.

27. The system according to claim 25, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (UT).

28. The system according to claim 25, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

29. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay D_1;

means for measuring, at said second object, a frequency from said first signal;

means for transmitting, from said second object to said first object, a second signal containing a report of the measured first signal frequency;

means for receiving, at said first object, said second signal after a propagation delay D_2; and

means for determining, at said first object, said round trip delay from said measured first signal frequency, said round trip delay being a function of the delay experienced during propagation of said second signal from said second object to said first object.

30. The system according to claim 29, wherein said frequency is (f_D-f_{offset}), where f_D is the Doppler frequency and f_{offset} is the second object's local oscillator error frequency.

31. The system according to claim 29, wherein said first object is a wireless telephone system orbiting satellite and said second object is a wireless telephone system user terminal (second object).

32. The system according to claim 29, wherein said first object is a wireless telephone system gateway and said second object is a wireless telephone system user terminal (UT).

33. A method for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

transmitting a first signal from said first object;

receiving said first signal at said second object after a propagation delay D_1;

transmitting a second signal from said second object to said first object upon receipt of said first signal;

receiving said second signal at said first object after a propagation delay D_2;

measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal;

measuring at said first object a frequency associated with said second signal; and

determining said round trip delay at said first object, said round trip delay being based upon (i) the measured frequency and (ii) the measured elapsed time.

34. A method for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

transmitting a first signal from said first object;

receiving said first signal at said second object after a propagation delay D_1;

measuring at said second object, a frequency associated with said first signal;

transmitting from said second object to said first object, a second signal containing a report of the measured first signal frequency;

receiving at said first object, said second signal after a propagation delay D_2;

measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal; and

determining said round trip delay at said first object, said round trip delay being based upon (i) the measured first signal frequency and (ii) the measured elapsed time.

35. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay D_1;

means for transmitting a second signal from said second object to said first object upon receipt by said second object of said first signal;

means for receiving said second signal at said first object after a propagation delay D_2;

means for measuring at said first object, a frequency associated with said second signal;

measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal; and

means for determining said round trip delay at said first object, said round trip delay being based upon (i) the measured frequency and (ii) the measured elapsed time.

36. A system for determining a round trip delay of signals transmitted between first and second objects that move relative to each other, comprising:

means for transmitting a first signal from said first object;

means for receiving said first signal at said second object after a propagation delay D_1;

means for measuring at said second object, a frequency associated with said first signal;

means for transmitting from said second object to said first object, a second signal containing a report of the measured first signal frequency;

means for receiving at said first object, said second signal after a propagation delay D_2;

measuring an elapsed time between the transmitting of the first signal and the receiving of the second signal; and

means for determining said round trip delay at said first object, said round trip delay being based upon (i) the measured first signal frequency and (ii) the measured elapsed time.

* * * * *

US006212171B1

(12) **United States Patent**
LaFollette et al.

(10) **Patent No.: US 6,212,171 B1**
(45) **Date of Patent: Apr. 3, 2001**

(54) **METHOD AND APPARATUS FOR GAP COUNT DETERMINATION**

(75) Inventors: **David LaFollette**, Sunnyvale; **Jerrold Hauck**, Fremont, both of CA (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/102,097**

(22) Filed: **Jun. 22, 1998**

(51) **Int. Cl.**^{7} **H04L 12/40**

(52) **U.S. Cl.** **370/257**; 370/252; 709/253; 710/104

(58) **Field of Search** 370/445, 448, 370/468, 231, 256, 437, 408, 257, 252; 709/253; 710/100, 104

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,504,757	*	4/1996	Cook et al.	370/468
5,687,319	*	11/1997	Cook et al.	370/256
5,802,057	*	9/1998	Duckwall et al.	370/408
5,991,520	*	11/1999	Smyers et al.	710/100

OTHER PUBLICATIONS

"Analysis of Gap Count Settings for the IEEE–1394 Bus"; prepared by Jim Skidmore; Texas Instruments Incorporated Mixed Signal Products Group. Feb. 26, 1997.
"P1394A Enhancments"; prepared by Bill Duckwall; available at "ftp//ftp.symbios.com", Jun. 3, 1997.

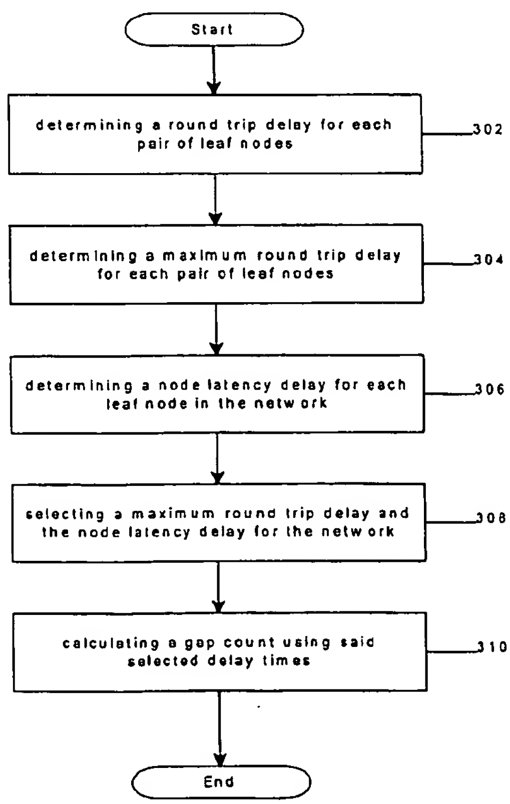
* cited by examiner

Primary Examiner—Hassan Kizou
Assistant Examiner—John Pezzlo
(74) *Attorney, Agent, or Firm*—Kenyon & Kenyon

(57) **ABSTRACT**

A method and apparatus for determining a gap count for a serial bus network is described. A round-trip delay time for transmitting a packet from a first leaf node to a second leaf node and back over a communication path between the nodes for each pair of leaf nodes in the network is determined. A maximum round-trip delay time for each communication path is calculated. A node latency delay time for each leaf node in the network is determined. A longest maximum round-trip delay time and a longest node latency delay time is selected for the network, and a gap count is calculated using the selected times.

31 Claims, 7 Drawing Sheets



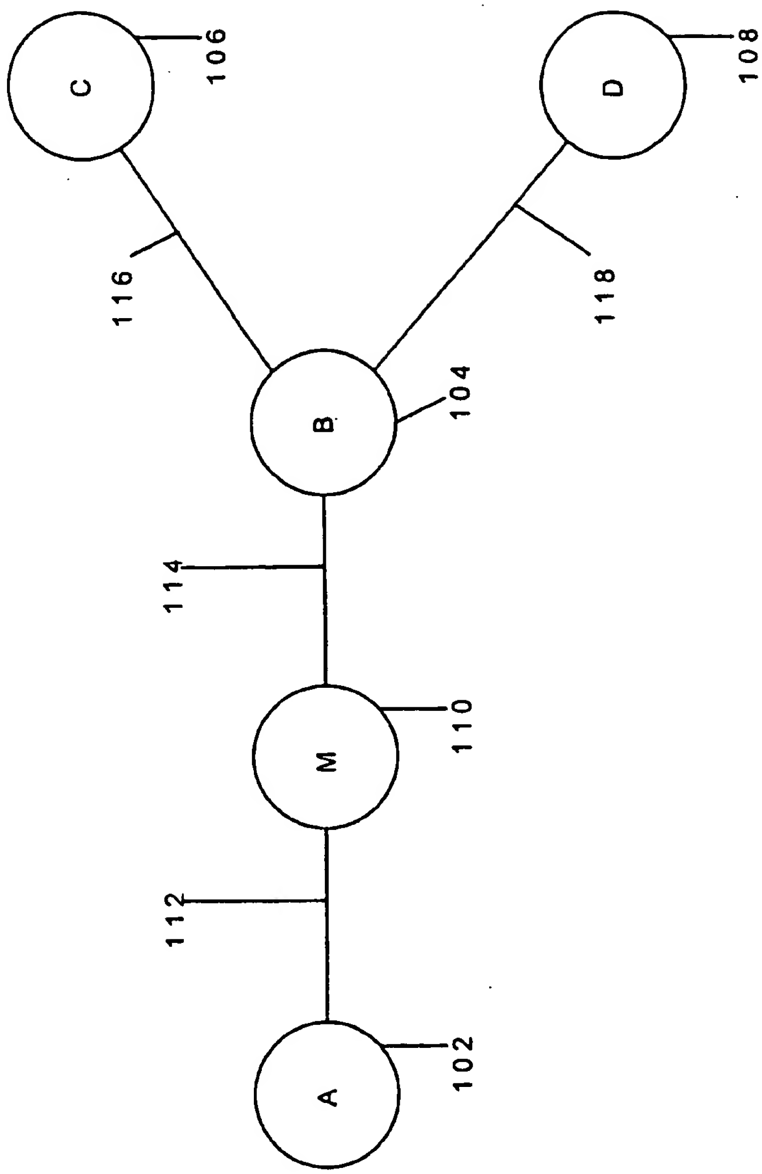


FIG. 1

100

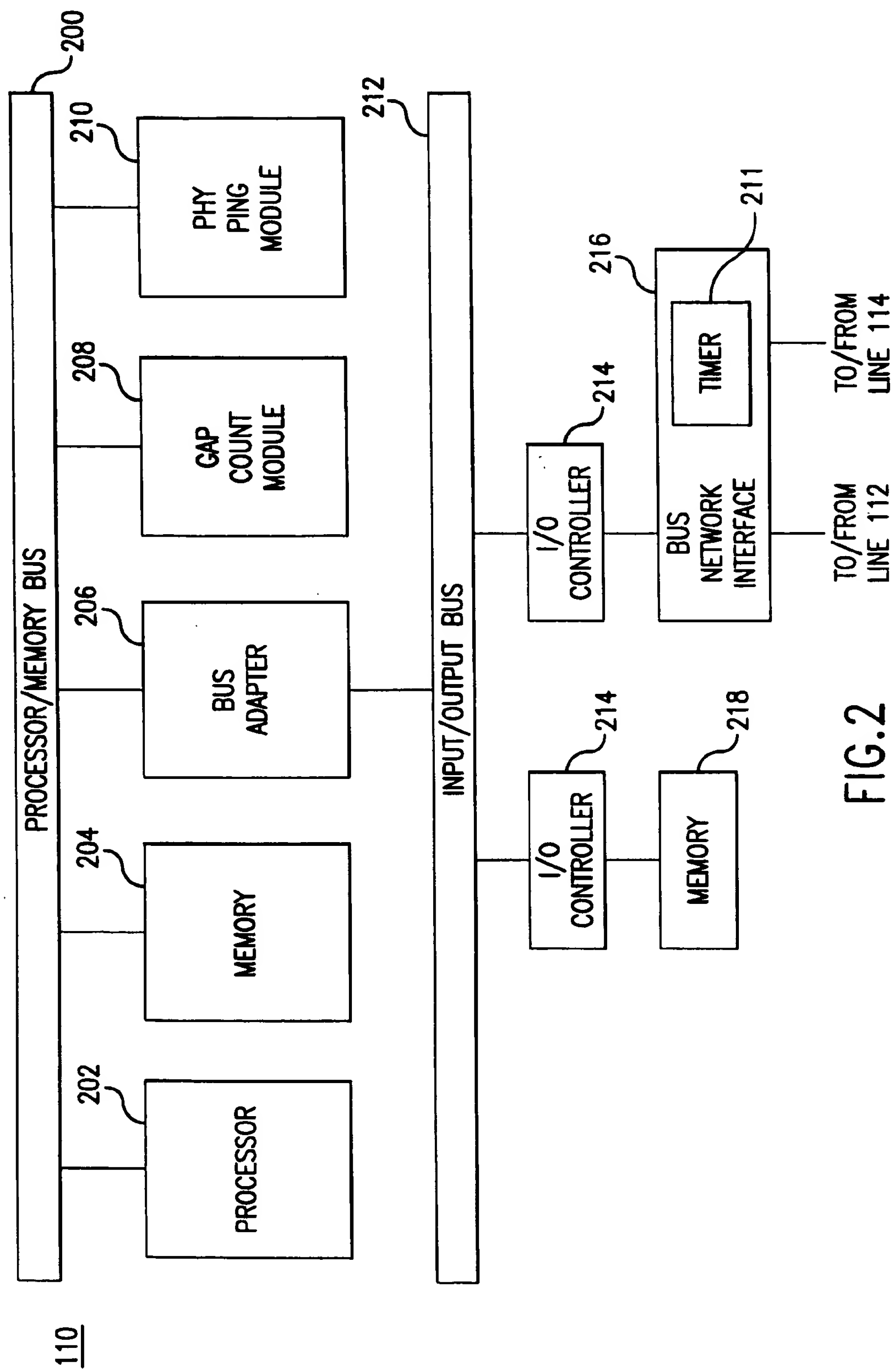

110
200
PROCESSOR/MEMORY BUS
202
PROCESSOR
204
MEMORY
206
BUS ADAPTER
208
GAP COUNT MODULE
210
PHY PING MODULE
212
INPUT/OUTPUT BUS
214
I/O CONTROLLER
214
I/O CONTROLLER
218
MEMORY
216
BUS NETWORK INTERFACE
211
TIMER
TO/FROM LINE 112
TO/FROM LINE 114

FIG.2

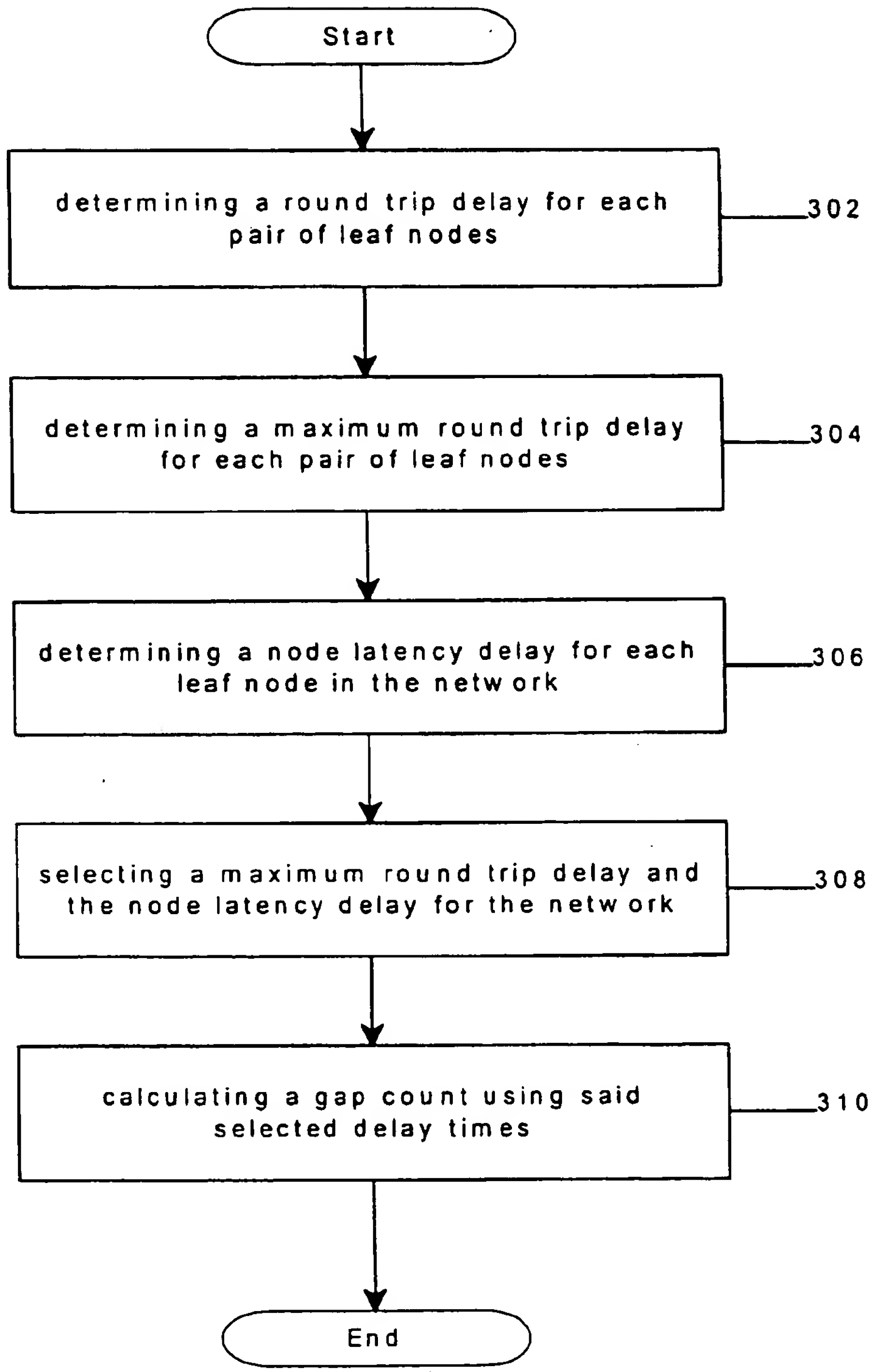

Start
determining a round trip delay for each pair of leaf nodes
302
determining a maximum round trip delay for each pair of leaf nodes
304
determining a node latency delay for each leaf node in the network
306
selecting a maximum round trip delay and the node latency delay for the network
308
calculating a gap count using said selected delay times
310
End

FIG. 3

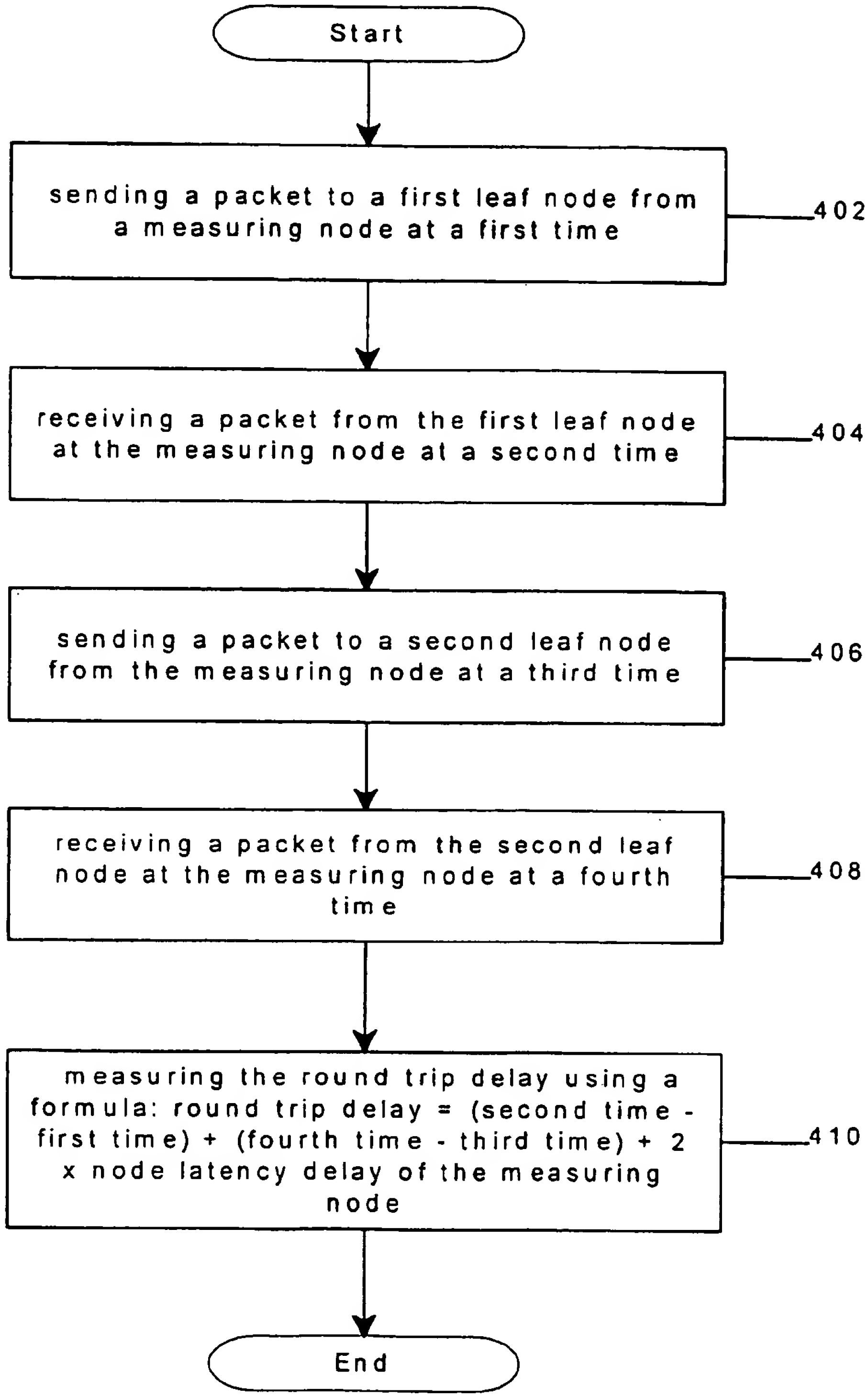

Start
sending a packet to a first leaf node from a measuring node at a first time
402
receiving a packet from the first leaf node at the measuring node at a second time
404
sending a packet to a second leaf node from the measuring node at a third time
406
receiving a packet from the second leaf node at the measuring node at a fourth time
408
measuring the round trip delay using a formula: round trip delay = (second time - first time) + (fourth time - third time) + 2 x node latency delay of the measuring node
410
End

FIG. 4

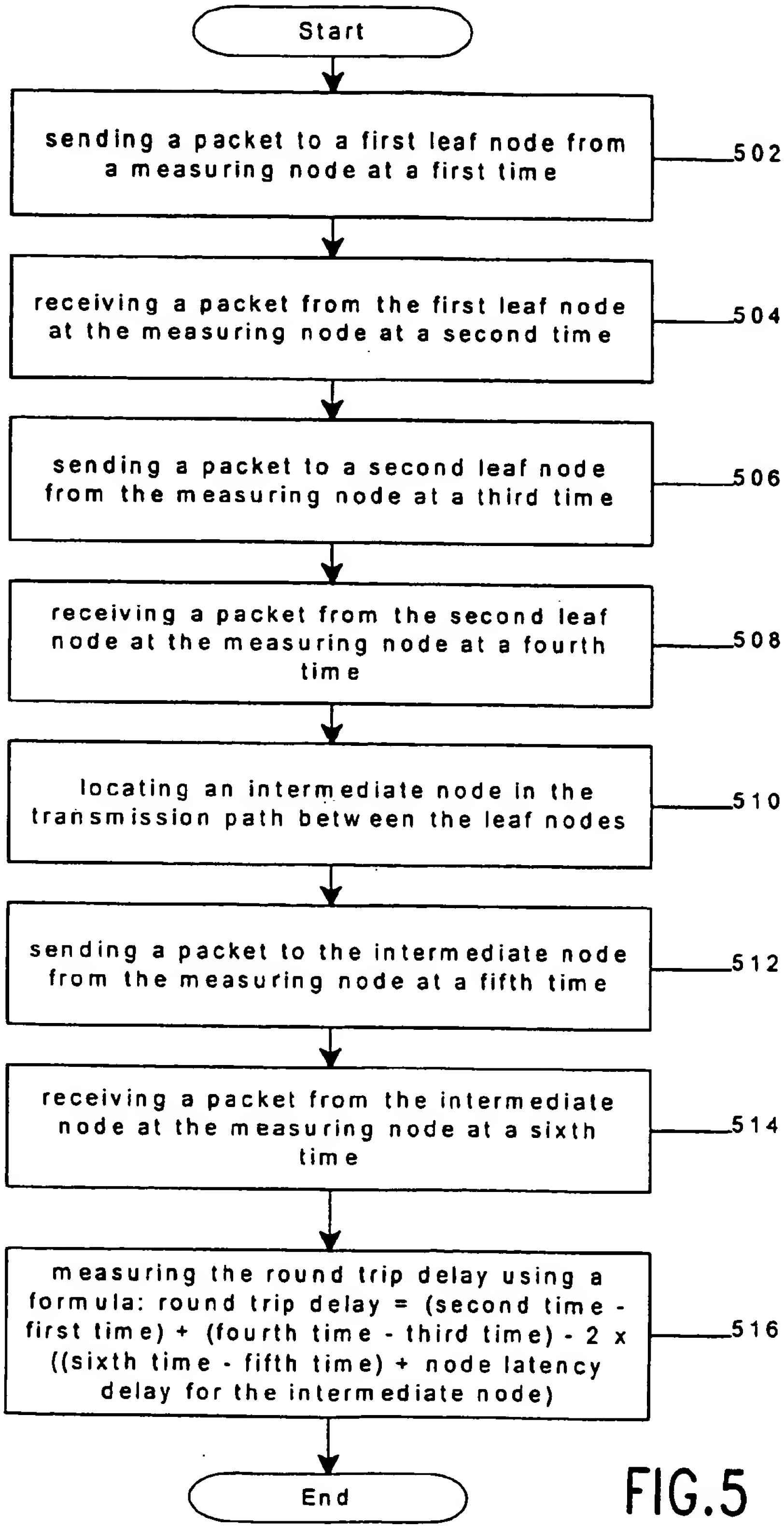

Start
sending a packet to a first leaf node from a measuring node at a first time
502
receiving a packet from the first leaf node at the measuring node at a second time
504
sending a packet to a second leaf node from the measuring node at a third time
506
receiving a packet from the second leaf node at the measuring node at a fourth time
508
locating an intermediate node in the transmission path between the leaf nodes
510
sending a packet to the intermediate node from the measuring node at a fifth time
512
receiving a packet from the intermediate node at the measuring node at a sixth time
514
measuring the round trip delay using a formula: round trip delay = (second time - first time) + (fourth time - third time) - 2 x ((sixth time - fifth time) + node latency delay for the intermediate node)
516
End

FIG.5

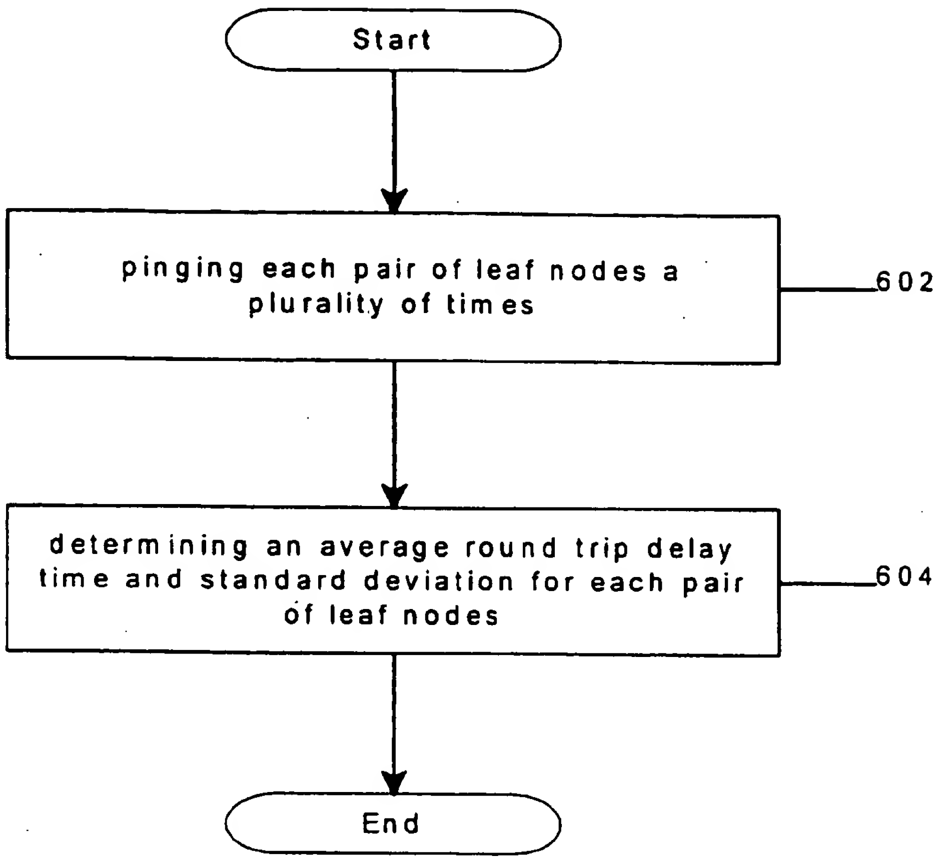

Start
pinging each pair of leaf nodes a plurality of times
602
determining an average round trip delay time and standard deviation for each pair of leaf nodes
604
End

FIG. 6

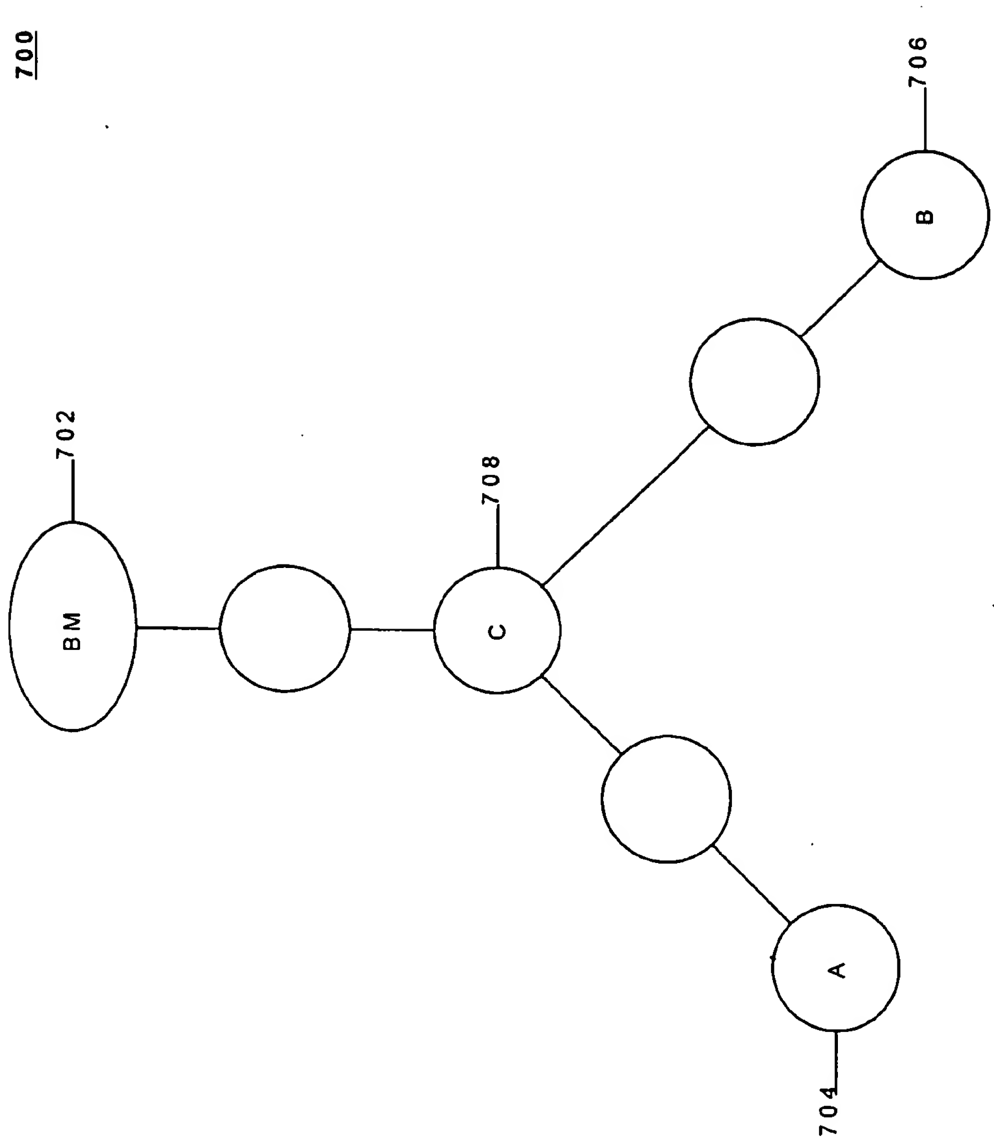


FIG. 7

METHOD AND APPARATUS FOR GAP COUNT DETERMINATION

FIELD OF THE INVENTION

The invention relates to bus interfaces in general. In particular, the invention relates to a method and apparatus for determining a gap count for a serial bus.

BACKGROUND OF THE INVENTION

The recognition of the superior quality of digital technologies has resulted in an unprecedented demand for digital products. This is evidenced by the popularity of consumer electronic devices such as audio compact discs, direct broadcast satellite systems, digital video disc (DVD) players, and digital video tape systems, as well as personal computer (PC) peripheral devices such as compact disc read-only memory (CD-ROM) drives, DVD drives, video cameras, musical instrument digital interface (MIDI) devices, and digital scanners.

The popularity of digital devices prompted a need for a uniform digital interface capable of connecting such devices into a single network. Consequently, the Institute of Electrical and Electronics Engineers (IEEE) introduced Standard 1394 titled "IEEE Standard for a High Performance Serial Bus," IEEE Computer Society, Dec. 12, 1995 ("IEEE 1394").

IEEE 1394 enables isochronous service while providing the bandwidth needed for audio, imaging, video, and other streaming data. Isochronous service means it guarantees latency (i.e., the length of time between a requested action and when the resulting action occurs). Latency is a critical feature in supporting real time video, for example. IEEE 1394 provides a high-speed serial bus with data transfer rates of 100, 200, or 400 mega-bits per second.

It is well-known in the art that IEEE 1394 data transfer rates can be improved by optimizing a parameter referred to as a "gap count." If the gap count is set too low, timing based arbitration breaks down. For instance, some nodes may detect idle time as an arbitration reset gap before others detect a subaction gap. The smaller the gap count is set, the smaller the bus topology must be for the timing to work. If the gap count is set too high, arbitration works, but the bus efficiency suffers from the larger than necessary gaps. Consequently, it is a design goal for IEEE 1394 network designers to set the gap count to the lowest workable value for a particular topology without interfering with the timing of the network.

One technique for determining an optimum gap count for a particular serial bus network is described in Annex E of IEEE 1394 itself. Table E-6 of Annex E lists predefined gap counts based upon the longest chain of nodes bounded by a pair of end nodes ("longest daisy chain") in the network, which is often measured in terms of "hops." For example, a serial bus network having seven nodes in its longest daisy-chain would have a total of six hops, which according to Table E-6 merits a gap count of six. Various techniques exist for determining the number of hops in a network. For example, "self-ID" packets sent by each node can be used to reconstruct the topology of a bus network, or the bus manager can simply assume a maximum number of hops (16) for a given network.

A gap count derived using Table E-6 and the various methods for determining hops, however, is unsatisfactory for at least one major reason. Of particular importance in optimizing a gap count is ascertaining a maximum round trip propagation delay for sending a packet of information from one end node to another end node for a given daisy-chain. A minimum gap count is then calculated using the longest round trip propagation delay found in the network, among various other factors. Table E-6 lists a maximum round trip propagation delay for a given number of hops based on the assumption that the length of the transmission medium (e.g., co-axial cable) between each node is 4.5 meters. The actual length of each cable, however, could range as high as 100 meters or more. Therefore, the maximum round trip propagation delay assumed by Table E-6 could in practice be far less than what is actually incurred in a network. As a result, the gap count derived using Table E-6 could be set low enough to cause timing problems throughout the network.

Bill Duckwall in a paper titled "1394a Enhancements" available at "ftp//ftp.symbios.com" (the "Duckwall reference") suggests calculating a gap count using a maximum round-trip delay measured by "PHY pinging." According to the Duckwall reference, a diagnostic node transmits a special packet containing a target address ("ping packet"). The node starts a timer when transmission is complete. When the target node receives the ping, it sends back a ping response packet. The diagnostic node detects the ping response, stops the timer at start of reception, and calculates from the timer the propagation delay to the target node. The diagnostic node selects the longest delay in the network and sets the gap count accordingly.

The technique disclosed in the Duckwall reference, however, is unsatisfactory for a number of reasons. Foremost, the Duckwall reference assumes the diagnostic node is actually one of the end nodes of the daisy chain having the longest delay in the network. This suggests that each end node must have the appropriate hardware and software to perform PHY pinging. This requirement is expensive and redundant. Further, the Duckwall reference assumes that the measured delay is the maximum propagation delay time. The measured delay, however, is merely representative of a single ping of a particular daisy-chain, and therefore may not represent the worst case delay over that particular daisy-chain without further information.

In view of the foregoing, it can be appreciated that a substantial need exists for a method and apparatus for determining an improved gap count for a given network topology that solves the above-discussed problems.

SUMMARY OF THE INVENTION

One embodiment of the invention comprises a method and apparatus for determining a gap count for a serial bus network. A round-trip delay time for transmitting a packet from a first leaf node to a second leaf node and back over a communication path between the nodes for each pair of leaf nodes in the network is determined. A maximum round-trip delay time for each communication path is calculated. A node latency delay time for each leaf node in the network is determined. A longest maximum round-trip delay time and a longest node latency delay time is selected for the network, and a gap count is calculated using the selected times.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of a serial bus network suitable for practicing one embodiment of the invention.

FIG. 2 is a block diagram of a measuring node in accordance with one embodiment of the invention.

FIG. 3 is a block flow diagram of the steps performed by a measuring node in accordance with one embodiment of the invention.

FIG. 4 is a block flow diagram of the steps for measuring round-trip delay times when the measuring node is on the path to be measured, in accordance with one embodiment of the invention.

FIG. 5 is a block flow diagram of the steps for measuring round-trip delay time when the measuring node is not on the path to be measured, in accordance with one embodiment of the invention.

FIG. 6 is a block flow diagram of the steps for determining a maximum round-trip delay for a pair of leaf nodes in accordance with one embodiment of the invention.

FIG. 7 is a serial bus network suitable for practicing another embodiment of the invention.

DETAILED DESCRIPTION

The various embodiments of the present invention comprise a method and apparatus for optimizing a gap count for a serial bus network, especially a network conforming to IEEE 1394. An optimal gap count significantly impacts the data transfer rates for a given network topology. If the gap count is set too low, timing based arbitration breaks down. For example, some nodes may detect idle time as an arbitration reset gap before others detect a subaction gap. The smaller the gap count is set, the smaller the bus topology must be for the timing to work. If the gap count is set too high, arbitration works, but the bus efficiency suffers from the larger than necessary gaps.

FIG. 1 is a block diagram of a serial bus network suitable for practicing one embodiment of the invention. It is worthy to note that any reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

FIG. 1 shows a serial bus network 100 that conforms to IEEE 1394. Network 100 comprises nodes A, B, C, D and M labeled 102, 104, 106, 108 and 110, respectively. Nodes A, M, B, C and D are connected by links 112, 114, 116 and 118, respectively. Node M (node 110) is a measuring node for network 100, and can be any node in network 100. In this embodiment of the invention, measuring node 110 is the bus manager node since the bus manager node, among other things, is responsible for optimizing performance of network 100.

As shown in FIG. 1, each node of network 100 is connected to another node via a communication link. In many instances, each node is connected to more than one node, such as nodes 104 and 110. There are, however, a number of nodes that are connected to only one other node in the network. Such nodes are typically referred to as "leaf nodes." For example, nodes 102, 106 and 108 are considered leaf nodes since they only connect with one other node. The importance of leaf nodes will be described in further detail later.

FIG. 2 is a block diagram of a measuring node in accordance with one embodiment of the invention. FIG. 2 shows measuring node 110 comprising a processor 202, a memory 204, a bus adapter 206, a gap count module 208, and a PHY ping module 210, each of which is connected to a processor/memory bus 200 and an Input/Output (I/O) bus 212 via bus adapter 206. Further, measuring node 110 contains a bus network interface 216 and memory 218, both of which are connected to I/O bus 212 via I/O controllers 214. The term "PHY" as used herein refers to the physical layer of a network as specified by IEEE 1394.

The overall functioning of measuring node 110 is controlled by processor 202, which operates under the control of executed computer program instructions and data stored in memory 204 or memory 218. Memory 204 may be any type of "fast" machine readable storage device, such as random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM), and so forth. Memory 218 may be any "slow" machine readable memory such as magnetic storage media (i.e., a magnetic disk), optical storage media (i.e., a CD-ROM or DVD), and so forth.

Processor 202 includes any processor of sufficient processing power to perform the functionality found in measuring node 110. Examples of processors suitable to practice the invention includes the Pentium™, Pentium™ Pro, and Pentium™ II microprocessors manufactured by Intel Corporation.

Bus network interface 216 controls communications between nodes via links 112, 114, 116 and 118 using the protocols, services and operating procedures set forth in IEEE 1394. Bus network interface 216 also includes a timer 211 for timing a round-trip delay time for packets sent between leaf nodes of a network. I/O controllers 214 control the flow of information between measuring node 110 and bus network interface 216 and memory 218. Bus adapter 206 transfers data back and forth between processor/memory bus 200 and I/O bus 212.

Gap count module 208 and PHY pinging module 210 implement the main functionality for measuring node 110. It is noted that modules 208 and 210 are shown as separate functional modules in FIG. 2. It can be appreciated, however, that the functions performed by these modules can be further separated into more modules, combined together to form one module, or be distributed throughout the system, and still fall within the scope of the invention. Further, the functionality of these modules may be implemented in hardware, software, or a combination of hardware and software, using well-known signal processing techniques. When implemented in software, the computer program segments or instructions are stored in computer readable memory such as memory 204 and memory 218, and are executed by a processor such as processor 202. The operation of these modules will be described in further detail below with reference to FIG. 3.

FIG. 3 is a block flow diagram of the steps performed by a measuring node in accordance with one embodiment of the invention. The communication path with the longest delay in the network is the one that determines the gap count. Since the actual delay across the physical layer for each node ("PHY delay" or "node latency delay")) and the actual delay across each cable connecting the nodes ("cable delays") are unknown, each leaf to leaf path must be checked and the worst one used. It is worthy to note that the longest delay is not necessarily in the path with the most nodes.

The measuring node measures a round trip delay time for a packet traveling over a communication path between a pair of leaf nodes for every pair of leaf nodes in the network at step 302. The measuring node measures round trip delay time by transmitting a ping packet and timing the return of a self-ID packet (IEEE P1394a) transmitted in response. Alternatively, any PHY packet that provokes a response from the addressed PHY may be used. For example, a remote access packet may be used both to obtain PHY jitter from another PHY and measure the propagation time in the

same step. Additional details with respect to step **302** will be discussed later with reference to FIGS. **4** and **5**.

Once the measuring node acquires a round trip delay time for a pair of leaf nodes, the measuring node uses the measured propagation times to calculate a maximum round trip time for each pair of leaf nodes at step **304**. This is because the measured propagation time changes for each measured ping, and therefore for purposes of setting a gap count cannot be considered the maximum delay time. Rather, it must be assumed to be the shortest time for this particular communication path. To set a gap count, however, a maximum round trip delay for the network must be determined. Therefore, a maximum round trip delay for each communication path must be first be determined. This can be accomplished in two ways, the first of which is described below. The second method is described with reference to FIG. **6** below.

The first method for determining a maximum round trip delay for a given communication path is through calculation. The time measured by PHY pinging from the measuring node to another node includes the data end of the PHY ping request, the delays through the measuring node, the cable and PHY propagation delays in both directions, the target arbitration response delay, the PHY ping response time and measurement error. This is represented by the following formula:

$$\mathrm{PingMeas}=\mathrm{DE}+\mathrm{LinkToPhyDel}_{BM}+2((\mathrm{hops}-1)^{*}(\mathrm{CD}+\mathrm{PD}))+2\mathrm{CD}+\mathrm{ArbRespDel}+\mathrm{PingRespTime}+\mathrm{PhyToLinkDel}_{BM}+\mathrm{MeasError}.$$

Since the measured propagation time consists essentially of cable delays and PHY delays, the maximum propagation time is calculated by adding jitter terms and subtracting the minimums of the undesired terms from the measured time. Jitter may be obtained from the PHY register 0100_2, as identified in IEEE 1394, for each repeating PHY along the path. Thus, the maximum round trip time is calculated using the following equation:

$$\mathrm{Propmax}=\mathrm{PingMeas}-\mathrm{DEMin}-\mathrm{LinkToPhyDelMin}_{BM}+2((\mathrm{hops}-1)^{*}(\mathrm{JPD}))-\mathrm{ArbRespDelMin}-\mathrm{PingRespTimeMin}-\mathrm{PhyToLinkDelMin}_{BM}.$$

The minimum propagation time is calculated by subtracting jitter terms and the minimums of the undesired terms from the measured time, as follows:

$$\mathrm{Propmin}=\mathrm{PingMeas}-\mathrm{DEMax}-\mathrm{LinkToPhyDelMax}_{BM}-2((\mathrm{hops}-1)^{*}(\mathrm{JPD}))-\mathrm{ArbRespDelMax}-\mathrm{PingRespTimeMax}-\mathrm{PhyToLinkDelMax}_{BM}.$$

The maximum and minimum round trip delay times for a given communication path are the aggregate cable and PHY delay adjusted for jitter. Any delay caused by arbitration or the PHY/link interface is subtracted out. The Ping Time, measured by link hardware, starts when the most significant bit of the ping packet is transferred from the link to the PHY and ends when a data prefix indication is signaled by the PHY. The term for PHY jitter is the sum of individual PHY jitter for each of the repeating PHYs on the path between the measuring node and the pinged node. This can be obtained by a remote read of the PHY registers. The resultant round-trip delay is expressed in units of microseconds.

In addition to determining maximum round trip delays for each communication path between leaf nodes in the network, the response time need for each leaf mode is determined at step **306**. This time is referred to as a leaf node latency delay or leaf PHY delay. Each leaf PHY delay can be determined from each leaf node's own PHY register.

As stated previously, the path having the worst (i.e., longest) round trip delay in the network is the one which determines the gap count. Therefore, the measuring node selects the value representing the longest maximum round-trip delay of all the leaf pairs in the network at step **308**. The worst leaf PHY delay is also selected at step **308**.

Using the worst maximum round-trip delay and PHY delay for the network, a gap count (GC) is calculated at step **310** using the following equation:

$$\mathrm{GC}=((\mathrm{BRMin})^{*}(\mathrm{BRMax})^{*}(\mathrm{RTDelMax}+\mathrm{ArbRespDelMax}_A+\mathrm{ArbRespDelMax}_B)-51\mathrm{BRMin}+29\mathrm{BRMax})/(32\mathrm{BRMin}-20\mathrm{BRMax}).$$

This formula can be reduced as follows:

$$\mathrm{GC}=8.196\times(\mathrm{RTDelMax}+\mathrm{ArbRespDelMax}_A+\mathrm{ArbRespDelMax}_B)-1.833.$$

The worst case round-trip delay is expressed in microseconds. The PHY delay term accounts for the maximum arbitration response delay of the two leaf nodes. The terms $\mathrm{ArbRespDelMax}_A$ and $\mathrm{ArbRespDelMax}_B$ may be replaced with the sum of the maximum PHY delays for the two leaf nodes for the worst case path. The resulting Gap Count is rounded to the next highest integer. The Gap Count may be transmitted in a PHY configuration packet to optimize the performance of network **100**.

As shown in FIG. **3** and described above, the measuring node determines a round-trip delay for each pair of leaf nodes in a network at step **302**. The technique used to determine the round-trip delay, however, varies according to where the measuring node is located with respect to the leaf nodes. The possible topologies resolve into three categories:

1. The measuring node is a leaf and the round-trip delay is to be measured to another leaf;
2. The measuring node is not a leaf but is on the path that connects two leaves whose round-trip delay is to be measured; and
3. The measuring node is neither a leaf nor on the path that connects two leaves whose round-trip delay is to be measured.

With respect to topology **1**, assume that nodes B, C and D are not present, that is, node M and node A are leaf nodes. The round-trip delay is the propagation time measured from node M to node A and back. Topologies **2** and **3** will be described below with reference to FIGS. **4** and **5**, respectively.

FIG. **4** is a block flow diagram of the steps for measuring round-trip delay times when the measuring node is on the path to be measured, in accordance with one embodiment of the invention. As shown in FIG. **4**, the measuring node sends a ping packet to a first leaf node from a measuring node at a first time at step **402**. A response packet is received at the measuring node from the first leaf node at step **404**. The measuring node sends a ping packet to a second leaf node from the measuring node at a third time at step **406**. A response packet is received at the measuring node from the second leaf node at a fourth time at step **408**. The round-trip delay for the path is calculated at step **410** using the following formula:

Round-trip delay=(second time−first time)+(fourth time−third time)+2xthe PHY delay of the measuring node.

For example, assume that the round-trip delay is to be calculated between nodes A and C as shown in FIG. **1**. Node M measures the propagation time from itself to node A and from itself to node C. The round-trip delay between the two leaf nodes also needs to account for the PHY delay of node M and is expressed by:

$$\text{Round-trip delay}_{(A,C)}=\text{Propagation time}_A+\text{Propagation time}_C+2\times\text{PHY delay}_M.$$

In the formula above, all of the times are maxima, and the measuring node's PHY delay is obtained from its own PHY registers.

FIG. **5** is a block flow diagram of the steps for measuring round-trip delay time when the measuring node is not on the path to be measured, in accordance with one embodiment of the invention. In this embodiment of the invention, steps **502**, **504**, **506** and **508** are similar to steps **402**, **404**, **406** and **408** described with reference to FIG. **4**, respectively. The measuring node then locates an intermediate node that is on the communication path being measured at step **510**. The measuring node sends a ping packet to the intermediate node at a fifth time at step **512**. The measuring node receives a response packet from the intermediate node at a sixth time at step **514**. The measuring node measures the round-trip delay at step **516** using the following formula:

$$\text{Round-trip delay}=(\text{second time}-\text{first time})+(\text{fourth time}-\text{third time})-2\times((\text{sixth time}-\text{fifth time})+\text{PHY delay for the intermediate node}).$$

Referring again to FIG. **1**, assume the measuring node is to determine a round-trip delay between nodes C and D. Node M measures propagation times to both nodes C and D. The measuring node then measures the propagation time to the node closest to the measuring node that is also on the path between the leaf nodes. In this example it would be node B. These measurements are combined to eliminate the propagation time from the measuring node to node B and the excess PHY delay for node B (measured twice in the propagation times for nodes C and D) as follows:

$$\text{Round-trip delay}_{(C,D)}=\text{Propagation time}_C+\text{Propagation time}_D-2\times(\text{Propagation time}_B+\text{PHY delaY}_B).$$

In this case, the propagation times measured for nodes C and D are minima while the propagation times measured to node B is maximum. The PHY delay for node B is obtained by remote access to that node's PHY registers.

Returning again to FIG. **3**, the measuring node determines a maximum round-trip delay for each pair of leaf nodes at step **304**. The maximum propagation time is calculated by adding jitter terms and subtracting the minimums of the undesired terms from the measured time. Alternatively, this can be accomplished as described with reference to FIG. **6** below.

FIG. **6** is a block flow diagram of the steps for determining a maximum round-trip delay for a pair of leaf nodes in accordance with one embodiment of the invention. The measuring node pings each pair of leaf nodes a plurality of times at step **602**. The measuring node then determines an average round-trip delay time and standard deviation for each pair of leaf nodes at step **604**. These values would represent actual jitter and are used to replace the calculated jitter terms described previously. This would tighten the bounds on the calculated propagation times and result in the selection of a more optimal gap count.

FIG. **7** is a bus serial network suitable for practicing one embodiment of the invention. FIG. **7** illustrates a network **700** comprising a bus manager node **702**, a node A (**704**), a node B (**706**) and a node C (**708**). Lines **710**, **712**, **714**, **716** and **720** are each 4.5 meters long. Line **718** is 100 meters long. Bus manager node **702** incorporates functionality similar to measuring node **110** as described with reference to FIGS. **1** and **2**.

A gap count will be determined for network **700** in accordance with the principles of the invention described above. The equations used to derive an optimal gap count are similar to those described with reference to FIGS. **1**–**6**, with some redefined terms, defined values, and suggested values, as shown in Table 1 below. In some instances, the term is a shortened version of a term defined in IEEE 1394 and IEEE P1394a. The subscripts BM, A, B and C refer to a Bus Manager and nodes A, B and C, respectively. It should be noted that the values listed in Table 1 are exemplary only, and can be modified in accordance with changes in the IEEE P1394a specification and still fall within the scope of the invention.

TABLE 1

TERM	IEEE 1394 MEANING	IEEE 1394 DEFINED VALUE	SUGGESTED VALUE
ArbDelMax	Maximum arb_delay	GC*4/BRMin	
ArbResetGapMin	Minimum generated Arb reset gap	(51 + GC * 32)/BRMax	
ArbRespDel			PD
BRMax	Maximum BASE_RATE	98.314 Mbit/S	
BRMin	Minimum BASE_RATE	98.294 Mbit/S	
CD	Cable Delay		
DE	DATA_END_TIME	0.24 uS to 0.26 uS	
GC	Gap Count	((BRMin)*(BRMax)*(RT DelMax + ArbRespDelMax_A + ArbRespDelMax_B) − 51 BRMin + 29 BRMax)/(32 BRMin − 20 BRMax)	
Hops	Cable hops between nodes		
JPD	Jitter in PHY Delay		20 nano-seconds (nS) (1 SClk period)
LinkToPhyDel	Link to PHY Delay		40 nS − 62 nS
MeasError	Phy Ping Measurement Error		
PD	PHY Delay	max 144 nS	min 33.3 nS

TABLE 1-continued

TERM	IEEE 1394 MEANING	IEEE 1394 DEFINED VALUE	SUGGESTED VALUE
PDDeltaMin		$PDMin_{BMA}$ + $PDMin_{BMB}$ - $PDMax_{AB}$	
PhyToLinkDel	PHY to Link Delay		81 nS - 102 nS (8/BRMax - 10/BRMin)
PingMeas	Measured round trip time to a node from measuring node	DE + $LinkToPhyDel_{BM}$ + 2((hops-1)*(CD + PD)) + 2 CD + ArbRespDel + PingRespTime + $PhyToLinkDel_{BM}$ + MeasError	
PingRespTime	PHY Ping Response Time		122 nS - 143 nS (12/BRMax - 14/BRMin)
PropMax	Calculated maximum round trip time to a node from measuring node	PingMeas - DEMin - $LinkToPhyDelMin_{BM}$ + 2((hops-1)*(JPD)) - ArbRespDelMin - PingRespTimeMin - $PhyToLinkDelMin_{BM}$	
PropMin	Calculated minimum round trip time to a node from measuring node	PingMeas - DEMax - $LinkToPhyDelMax_{BM}$ - 2((hops-1)*(JPD)) - ArbRespDelMax - PingRespTimeMax - $PhyToLinkDelMax_{BM}$	
RTDel	Round Trip Delay	2(hops-1)*(CD+PD) + 2 CD	
SubactionGapMax	Maximum Observed Subaction Gap	(29 + 16 GC)/BRMin + ArbDelMax + RTDelMax + $ArbRespDelMax_A$ + $ArbRespDelMax_B$	

The following assumptions are made in calculating a gap count for network **700**. A Round Trip delay is defined to include round trip cable and PHY propagation delays as follows: RTDel=2(hops−1)*(CD+PD)+2CD. To insure that no node sees an arb reset gap before another node sees a subaction gap, Gap Count is set such that ArbResetGapMin>SubactionGapMax. The minimum detection time for arb reset gap is as follows: ArbResetGapMin=(51+32GC)/BRMax. The maximum observed subaction gap is as follows: SubactionGapMax=(29+16GC)/BRMin+ArbDelMax+RTDelMax+$ArbRespDelMax_A$+$ArbRespDelMax_B$, with ArbDelMax=4GC/BRMin. Thus, for the smallest usable gap count:

$$(51+32GC)/BRMax=(29+16GC)/BRMin+ArbDelMax+RTDelMax+ArbRespDelMax_A+ArbRespDelMax_B;$$

$$51BRMin+32BRMin(GC)=29BRMax+16BRMax(GC)+4BRMax(GC)+(BRMin)*(BRMax)*(RTDelMax+ArbRespDelMax_A+ArbRespDelMax_B);$$

$$GC(32BRMin-20BRMax)=(BRMin)*(BRMax)*(RTDelMax+ArbRespDelMax_A+ArbRespDelMax_B)-51BRMin+29BRMax;$$

$$GC=((BRMin)*(BRMax)*(RTDelMax+ArbRespDelMax_A+ArbRespDelMax_B)-51BRMin+29BRMax)/(32BRMin-20BRMax).$$

The following assumptions are made in determining a maximum round trip delay for network **700**. The time measured by PHY pinging from the Bus Manager to another node is shown by PingMeas=DE+$LinkToPhyDel_{BM}$+2((hops−1)*(CD+PD))+2CD+ArbRespDel+PingRespTime+$PhyToLinkDel_{BM}$+MeasError. The calculated maximum propagation time is the measured time plus uncertainties minus the minimums of the undesired terms, as shown in PropMax=PingMeas−DEMin−$LinkToPhyDelMin_{BM}$+2((hops−1)*(JPD))−ArbRespDelMin−PingRespTimeMin−$PhyToLinkDelMin_{BM}$. The calculated minimum propagation time is the measured time minus uncertainties minus the maximums of the undesired terms, as shown in PropMin=PingMeas−DEMax−$LinkToPhyDelMaX_{BM}$−2((hops−1)*(JPD))−ArbRespDelMax−PingRespTimeMax−$PhyToLinkDelMax_{BM}$.

When pinging from the bus manager node to nodes A and B, the PHY delays seen through node C are along the paths between the port leading to the bus manager and the ports leading to nodes A and B. The PHY delay of interest, however, is the one between the child ports along the path from node A to node B. Thus, the PHY Delay Delta is defined as the difference between the sum of the port to port paths between the bus manager node and nodes A and B minus the port to port path between nodes A and B. It is worthy to note that this value may be different in nodes which have 1394B ports than in 1394A nodes.

In this example, the bus manager must ping all the other leaf nodes, A and B. It must then calculate the maximum round trip delay for all leaf to leaf paths, BM to A, BM to B, and A to B. The biggest round trip delay is then used to calculate a gap count for network **700**.

The following additional values are defined for network **700** in Table 2 below.

TABLE 2

Term	Actual Value
DE	250nS
$LinkToPhyDel_{BM}$	51nS
CD	5.05nS/M*length
PD	100nS
$ArbRespDel_A$	100nS

TABLE 2-continued

Term	Actual Value
PingRespTime	132nS
$PhyToLinkDel_{BM}$	92nS
MeasError	10nS

The measured round trip time to nodes A, B and C from the bus manager are:

$$
\begin{aligned}
PingMeas_A &= DE + LinkToPhyDel_{BM} + 2((Hops-1)(CD+PD)) + 2{*}CD + ArbRespDel_A + PingRespTime_A + PhyToLinkDel_{BM} + MeasError \\
&= 250\ nS + 51\ nS + 8{*}5.05\ nS/M{*}4.5\ M + 6{*}100\ nS + 100\ nS + 132\ nS + 92\ nS + 10\ nS \\
&= 1417\ nS \\
PingMeas_B &= DE + LinkToPhyDel_{BM} + 2((Hops-1)(CD+PD)) + 2{*}CD + ArbRespDel_B + PingRespTime_B + PhyToLinkDel_{BM} + MeasError \\
&= 250\ nS + 51\ nS + 6{*}5.05\ nS/M{*}4.5\ M + 2{*}5.05\ nS/M{*}100\ m + 6 * 100\ nS + 100\ nS + 132\ nS + 92\ nS + 10\ nS \\
&= 2381\ nS \\
PingMeas_C &= DE + LinkToPhyDel_{BM} + 2((Hops-1)(CD+PD)) + 2{*}CD + ArbRespDel_C + PingRespTime_C + PhyToLinklDel_{BM} + MeasError \\
&= 250\ nS + 51\ nS + 4 * 5.05\ nS/M * 4.5\ M + 2{*}100\ nS + 100\ nS + 132\ nS + 92\ nS + 10\ nS \\
&= 926\ nS
\end{aligned}
$$

The following equations do not assume the use of statistical methods to eliminate the Phy delay jitter terms. Results may be improved if nodes are pinged repeatedly and statistical methods are used as described with reference to FIG. 6.

$$
\begin{aligned}
PropMax_A &= PingMeas_A - DEMin - LinkToPhyDelMin_{BM} + 2((hops-1)(JPD)) - ArbRespDelMin - PingRespTimeMin - PhyToLinkDelMin_{BM} \\
&= 1417\ nS - 240\ nS - 40\ nS + 6{*}20\ nS - 33.3\ nS - 122\ nS - 81\ nS \\
&= 1021\ nS \\
PropMax_B &= PingMeas_B - DEMin - LinkToPhyDelMin_{BM} + 2((hops-1)(JPD)) - ArbRespDelMin - PingRespTimeMin - PhyToLinkDelMin_{BM} \\
&= 2381\ nS - 240\ nS - 40\ nS + 6{*}20\ nS - 33.3\ nS - 122\ nS - 81\ nS \\
&= 1985\ nS \\
PropMin_C &= PingMeas_C - DEMax - LinkToPhyDelMax_{BM} - 2((hops-1)(JPD)) - ArbRespDelMax - PingRespTimeMax - PhyToLinkDelMax_{BM} \\
&= 926\ ns - 260\ nS - 61\ ns - 2{*}20\ nS - 144\ nS - 142\ nS - 102\ nS \\
&= 177\ nS
\end{aligned}
$$

The Maximum Round Trip Time Calculations are as follows:

$$
\begin{aligned}
RTDelMax_{BMA} &= PropMax_A \\
&= 1021\ nS \\
RTDelMax_{BMB} &= PropMax_B \\
&= 1985\ nS \\
PDDeltaMin &= PDMin_{BMA} + PDMin_{BMB} - PDMax_{AB} \\
&= 33.3\ nS + 33.3\ nS - 144\ nS \\
&= -77.4\ nS
\end{aligned}
$$

-continued

$$
\begin{aligned}
RTDelMax_{AB} &= PropMax_A + PropMax_B - 2(PropMin_C) - 2(PDMin_C) - 2(PDDeltaMin_C) \\
&= 1021\ nS + 1985\ nS - 2(177\ nS) - 2(33.3\ nS) - 2(-77.4\ nS) \\
&= 2740\ nS
\end{aligned}
$$

The largest round trip delay calculated for any leaf to leaf path is 2740 nS for node A to node B. Gap count is calculated using this value as follows:

$$
\begin{aligned}
GC &= ((BRMin)(BRMax)(RTDelMax_{AB} + ArbRespDelMax_A + ArbRespDelMax_B) - 51\ BRMin + 29\ BRMax)/(32\ BRMin - 20\ BRMax) \\
&= ((98.294)(98.314)(2.74\ uS + 0.144\ uS + 0.144\ uS) - 51(98.294) + 29{*}(98.314))/(32(98.294) - 20{*}(98.314)) \\
&= 22.98
\end{aligned}
$$

The gap count in this example must be set to 23 to guarantee node A never sees an arbitration reset gap before node B sees a subaction gap.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, although network **100** comprises only five nodes, it can be appreciated that a serial bus network can comprise any number of nodes in an infinite variety of patterns and still fall within the scope of the invention. For another example, although a PHY ping packet was used to measure propagation delays for network **100**, it can be appreciated that any type of timing packet falls within the scope of the invention.

What is claimed is:

1. A method for determining a gap count for a serial bus network, comprising:

determining a round-trip delay time for transmitting a packet from a first leaf node to a second leaf node and back over a communication path between said nodes for each pair of leaf nodes in the network;

calculating a maximum round-trip delay time for each communication path;

determining a node latency delay time for each leaf node in the network;

selecting a longest maximum round-trip delay time and a longest node latency delay time for the network; and

calculating a gap count using said selected times.

2. The method of claim 1, wherein said determining is performed by a measuring node.

3. The method of claim 1, wherein said determining a round-trip delay comprises:

pinging each pair of leaf nodes a plurality of times; and

determining an average round-trip delay time and standard deviation for each pair of leaf nodes.

4. The method of claim 2, wherein said measuring node is one of said first and second leaf nodes, and wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said second leaf node at a first time;

receiving a packet from said first leaf node at said second leaf node at a second time; and

measuring said round-trip delay using said first and second times.

5. The method of claim 2, wherein said measuring node is a node other than said first and second leaf nodes, and wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said measuring node at a first time;

receiving a packet from said first leaf node at said measuring node at a second time;

sending a packet to said second leaf node from said measuring node at a third time;

receiving a packet from said second leaf node at said measuring node at a fourth time; and

measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)+2×node latency delay of said measuring node.

6. The method of claim 2, wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said measuring node at a first time;

receiving a packet from said first leaf node at said measuring node at a second time;

sending a packet to said second leaf node from said measuring node at a third time;

receiving a packet from said second leaf node at said measuring node at a fourth time;

locating an intermediate node in said communication path between said leaf nodes;

sending a packet to said intermediate node from said measuring node at a fifth time;

receiving a packet from said intermediate node at said measuring node at a sixth time; and

measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)−2×((sixth time−fifth time)+node latency delay for said intermediate node).

7. The method of claim 1, wherein said calculating uses a formula:

Gap Count=8.196×(maximum round-trip delay time+2×maximum node latency delay)−1.833.

8. The method of claim 6, further comprising rounding said gap count to a next largest integer.

9. An apparatus for determining a gap count for a serial bus network, comprising:

means for determining a round-trip delay time for transmitting a packet from a first leaf node to a second leaf node and back over a communication path between said nodes for each pair of leaf nodes in the network;

means for calculating a maximum round-trip delay time for each communication path;

means for determining a node latency delay time for each leaf node in the network;

means for selecting a longest maximum round-trip delay time and a longest node latency delay time for the network; and

means for calculating a gap count using said selected times.

10. The apparatus of claim 9, wherein said means for determining a round-trip delay time is located in a measuring node.

11. The apparatus of claim 9, wherein said means for determining a round-trip delay comprises:

means for pinging each pair of leaf nodes a plurality of times; and

means for determining an average round-trip delay time and standard deviation for each pair of leaf nodes.

12. The apparatus of claim 10, wherein said measuring node is one of said first and second leaf nodes, and wherein said means for determining a round-trip delay comprises:

means for sending a packet to said first leaf node from said second leaf node at a first time;

means for receiving a packet from said first leaf node at said second leaf node at a second time; and

means for measuring said round-trip delay using said first and second times.

13. The apparatus of claim 10, wherein said measuring node is a node other than said first and second leaf nodes, and wherein said means for determining a round-trip delay comprises:

means for sending a packet to said first leaf node from said measuring node at a first time;

means for receiving a packet from said first leaf node at said measuring node at a second time;

means for sending a packet to said second leaf node from said measuring node at a third time;

means for receiving a packet from said second leaf node at said measuring node at a fourth time; and

means for measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)+2×node latency delay of said measuring node.

14. The apparatus of claim 10, wherein said means for determining a round-trip delay comprises:

means for sending a packet to said first leaf node from said measuring node at a first time;

means for receiving a packet from said first leaf node at said measuring node at a second time;

means for sending a packet to said second leaf node from said measuring node at a third time;

means for receiving a packet from said second leaf node at said measuring node at a fourth time;

means for locating an intermediate node in said communication path between said leaf nodes;

means for sending a packet to said intermediate node from said measuring node at a fifth time;

means for receiving a packet from said intermediate node at said measuring node at a sixth time; and

means for measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)−2×((sixth time−fifth time)+node latency delay for said intermediate node).

15. The apparatus of claim 9, wherein said means for calculating uses a formula:

Gap Count=8.196×(maximum round-trip delay time+2×maximum node latency delay)−1.833.

16. The apparatus of claim 15, further comprising a means for rounding said gap count to a next largest integer.

17. A computer-readable medium whose contents cause a computer system to determine a gap count for a serial bus network, by:

determining a round-trip delay time for transmitting a packet from a first leaf node to a second leaf node and

back over a communication path between said nodes for each pair of leaf nodes in the network;

calculating a maximum round-trip delay time for each communication path;

determining a node latency delay time for each leaf node in the network;

selecting a longest maximum round-trip delay time and a longest node latency delay time for the network; and

calculating a gap count using said selected times.

18. The computer-readable medium of claim 17, wherein said determining is performed by a measuring node.

19. The computer-readable medium of claim 17, wherein said determining a round-trip delay comprises:

pinging each pair of leaf nodes a plurality of times; and

determining an average round-trip delay time and standard deviation for each pair of leaf nodes.

20. The computer-readable medium of claim 18, wherein said measuring node is one of said first and second leaf nodes, and wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said second leaf node at a first time;

receiving a packet from said first leaf node at said second leaf node at a second time; and

measuring said round-trip delay using said first and second times.

21. The computer-readable medium of claim 18, wherein said measuring node is a node other than said first and second leaf nodes, and wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said measuring node at a first time;

receiving a packet from said first leaf node at said measuring node at a second time;

sending a packet to said second leaf node from said measuring node at a third time;

receiving a packet from said second leaf node at said measuring node at a fourth time; and

measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)+2×node latency delay of said measuring node.

22. The computer-readable medium of claim 18, wherein said determining a round-trip delay comprises:

sending a packet to said first leaf node from said measuring node at a first time;

receiving a packet from said first leaf node at said measuring node at a second time;

sending a packet to said second leaf node from said measuring node at a third time;

receiving a packet from said second leaf node at said measuring node at a fourth time;

locating an intermediate node in said communication path between said leaf nodes;

sending a packet to said intermediate node from said measuring node at a fifth time;

receiving a packet from said intermediate node at said measuring node at a sixth time; and

measuring said round-trip delay using a formula:

round-trip delay=(second time−first time)+(fourth time−third time)−2×((sixth time−fifth time)+node latency delay for said intermediate node).

23. The computer-readable medium of claim 17, wherein said calculating uses a formula:

Gap Count=8.196×(maximum round-trip delay time+2×maximum node latency delay)−1.833.

24. The computer-readable medium of claim 23, further comprising rounding said gap count to a next largest integer.

25. A computer for determining a gap count for a serial bus network, comprising: a memory containing:

a round-trip delay time module;

a maximum round-trip delay time module;

a node latency delay time module;

a selection module;

a gap count module; and

a processor for executing said modules.

26. A measuring node for determining a gap count for a serial bus network, comprising:

a pinging module having an output for determining a round-trip delay time;

a timer coupled to said pinging module, wherein said output is sent to a plurality of nodes and said timer measures the round-trip delay time between each pair of said plurality nodes; and

a gap count determination circuit having an input coupled to said output for receiving said round-trip delay time between each pair of nodes and calculating said gap count for the network and wherein said gap count determination circuit determines a node latency delay time across said plurality of nodes.

27. An apparatus determining a gap count for a serial bus network, comprising:

at least one pair of leaf nodes;

a communication link connecting said leaf nodes; and

a measuring node in communication with said leaf nodes, wherein said measuring node determines a round trip delay time between said leaf nodes over said communication path for use in determining the gap count and said measuring node determines a node latency delay time across said leaf nodes.

28. The measuring node of claim 26, wherein said timer for further determining an average round trip delay time and a standard deviation for said any pair of nodes in the network.

29. The apparatus of claim 27, wherein said measuring node is a node other than said at least one pair of leaf nodes.

30. The apparatus of claim 27, wherein said measuring node is located in said communication link connecting said leaf nodes.

31. The apparatus of claim 27, wherein said measuring node is located outside said communication link connecting said leaf nodes.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,212,171 B1
DATED : April 3, 2001
INVENTOR(S) : LaFollette et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2,
Line 57, delete "is" and insert -- are --;

Column 7,
Line 35, delete "delaY_B" and insert -- $delay_B$ --;

Column 16,
Line 36, after "network" insert -- , --.

Signed and Sealed this

Twenty-eighth Day of May, 2002

Attest:

JAMES E. ROGAN

Attesting Officer

Director of the United States Patent and Trademark Office

US005864540A

United States Patent [19]

Bonomi et al.

[11] **Patent Number: 5,864,540**

[45] **Date of Patent: Jan. 26, 1999**

[54] **METHOD FOR INTEGRATED TRAFFIC SHAPING IN A PACKET-SWITCHED NETWORK**

[75] Inventors: **Flavio Giovanni Bonomi**, Palo Alto, Calif.; **Albert Gordon Greenberg**, Millburn; **Jennifer Lynn Rexford**, Summit, both of N.J.

[73] Assignee: **AT&T Corp/CSI Zeinet(A Cabletron Co.)**, Middletown, N.J.

[21] Appl. No.: **825,990**

[22] Filed: **Apr. 4, 1997**

[51] **Int. Cl.**[6] **H04J 3/14**

[52] **U.S. Cl.** **370/235**; 370/413

[58] **Field of Search** 370/229, 230, 370/231, 232, 233, 234, 235, 236, 412, 413, 414, 415, 416, 417, 418, 389, 392, 395, 428, 429, 465, 468

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,260,935	11/1993	Turner	370/418
5,268,900	12/1993	Hluchyi	370/418
5,499,238	3/1996	Shon	370/399
5,515,363	5/1996	Ben-Nun et al.	370/412
5,602,830	2/1997	Fichou et al.	370/412
5,629,937	5/1997	Hayter et al.	370/412

OTHER PUBLICATIONS

Pierre E. Boyer, Fabrice M. Buillemin, Michael J. Servel, and Jean–Pierre Courdreuse; "Spacing Cells Protects and Enhances Utilization of ATM Network Links," IEEE Network, Sep., 1992, pp. 38–49.

Eugene Wallmeir and Tom Worster; "The Spacing Policer, An Algorithm for Efficient Peak Bit Rate Control in ATM Networks," Proceedings of 14th International Switching Symposium; Oct., 1992, vol. 2, A5.5, pp. 22–26.

Jennifer L. Rexford, Albert G. Greenberg, and Flavio G. Bonomi; "Hardware–Efficient Fair Queueing Architectures for High–Speed Networks," Proceedings of IEEE Infocom '96, Mar., 1996.

Jon C. R. Bennet and Hui–Zhang, "Hierarchical Packet Fair Queueing Algorithms," Proceedings of ACM SIGCOMM, Aug., 1996, pp. 143–156.

Jennifer Rexford, Flavio Bonomi, Albert Greenberg, and Albert Wong; IEEE INFOCOM; A Scalable Architecture for Fair Leaky–Bucket Shaping, 1997.

Jennifer Rexford, Flavio Bonomi, Albert Greemberg and Alberg Wong; IEEE Journal on Selected Areas in Communications Special Issue on Advances in ATM Switching Systems; Scalable Architectures for integrated Traffic Shaping and Link Scheduling in High–Speed ATM Switches, 1997.

Primary Examiner—Huy D. Vu

[57] **ABSTRACT**

A scalable integrated traffic shaper for a use in a packet-switched network that regulates multiple connections and prevents lost data by integrating link scheduling and traffic shaping to fairly arbitrate between incoming connections.

18 Claims, 5 Drawing Sheets

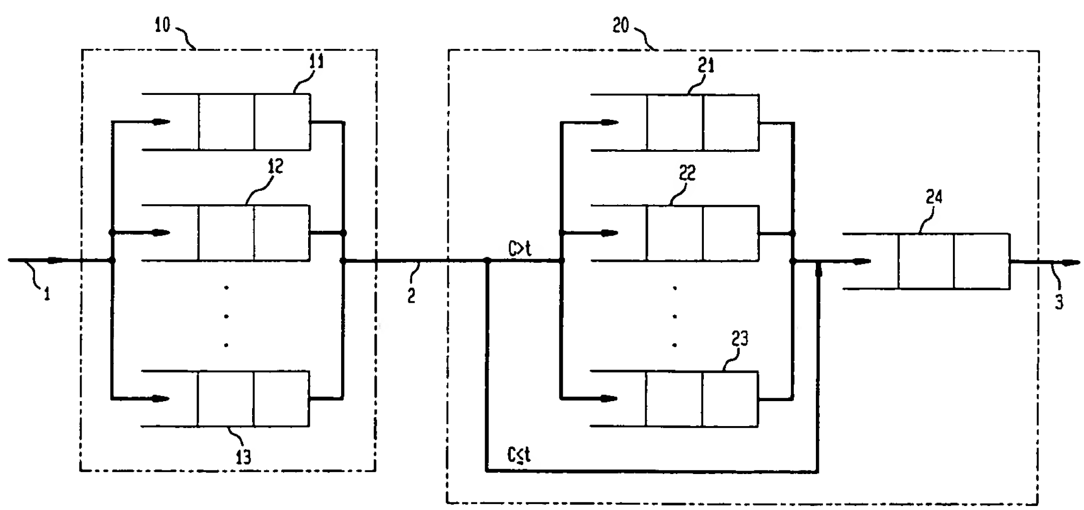


FIG. 1

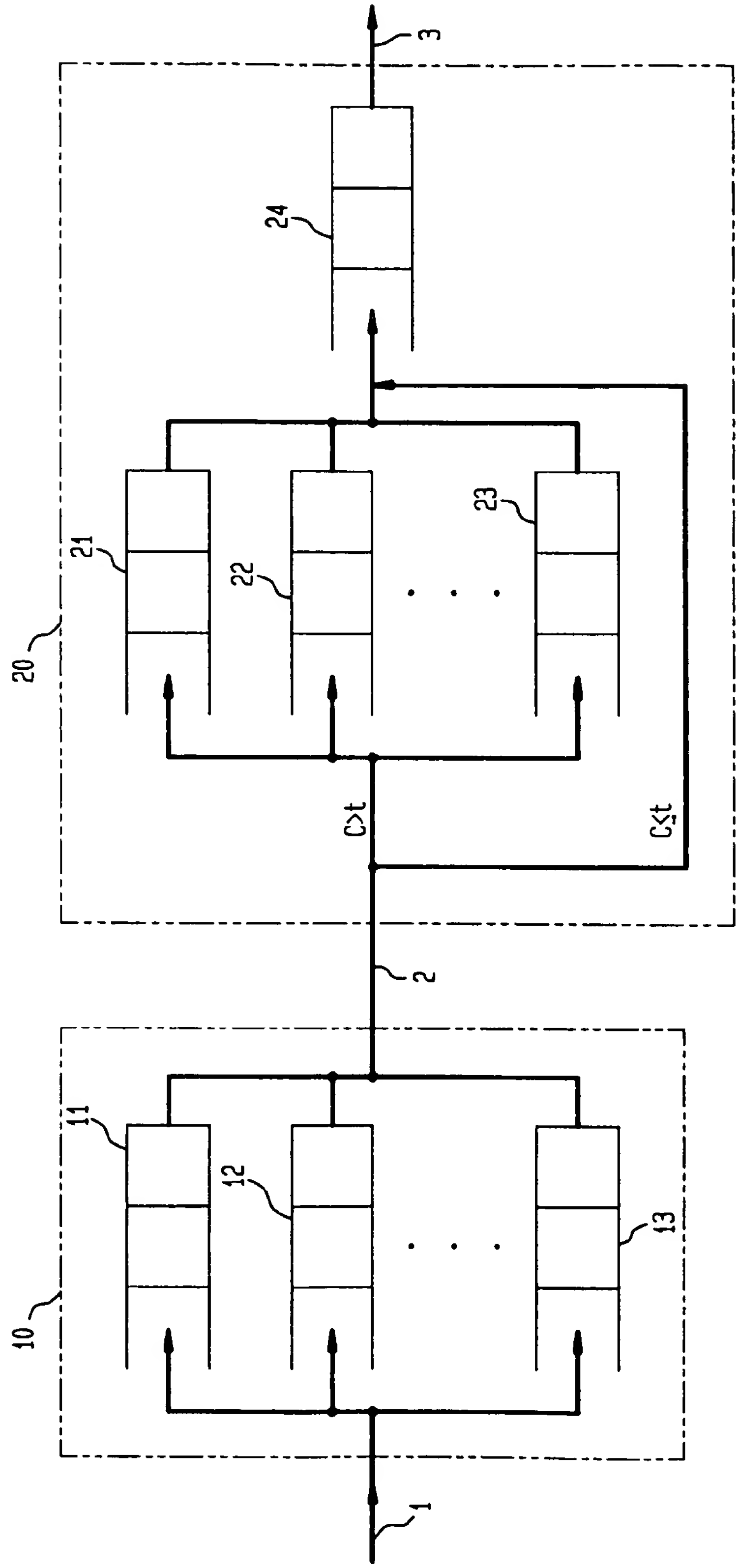


FIG. 2

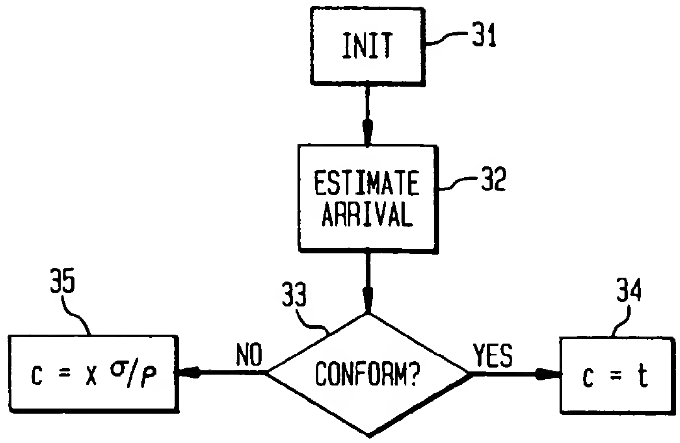

INIT
31
ESTIMATE ARRIVAL
32
35
33
34
c = x σ/ρ
NO
CONFORM?
YES
c = t

FIG. 3

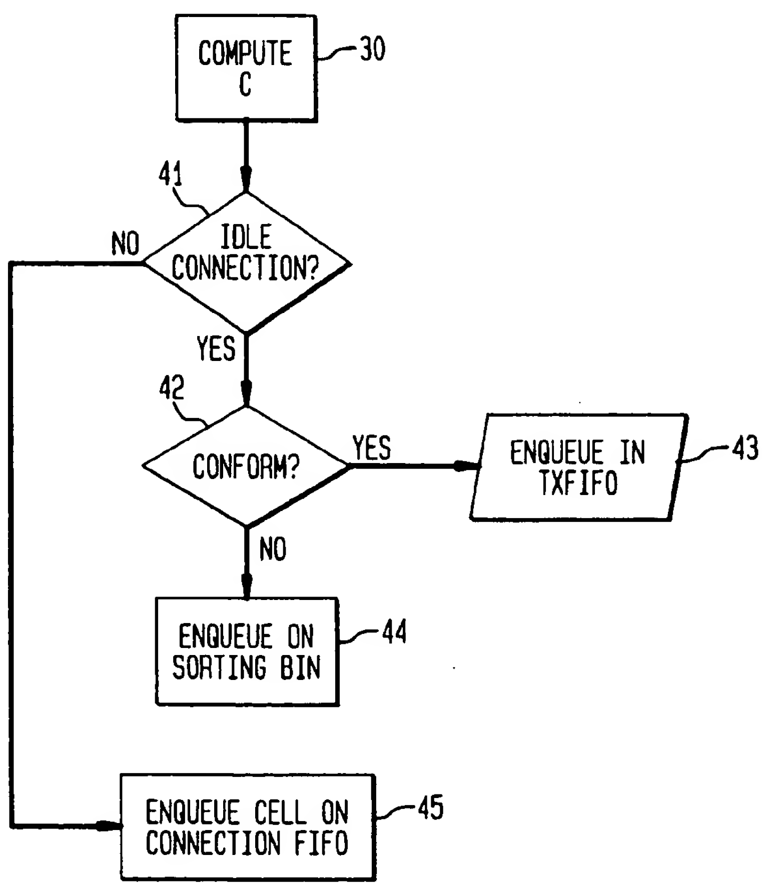
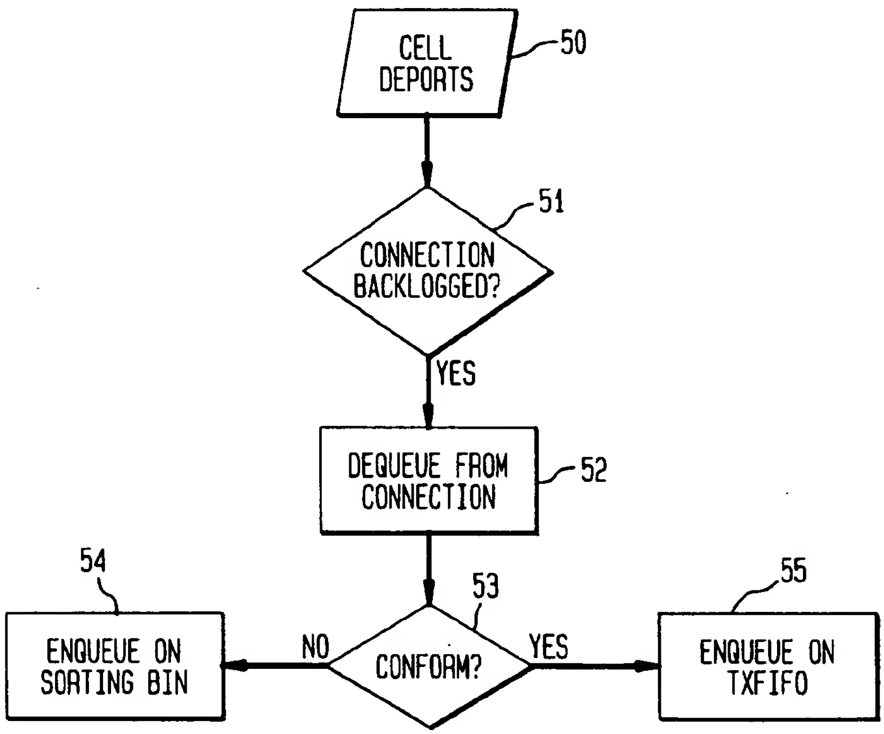

COMPUTE C
30
41
NO
IDLE CONNECTION?
YES
42
CONFORM?
YES
ENQUEUE IN TXFIFO
43
NO
ENQUEUE ON SORTING BIN
44
ENQUEUE CELL ON CONNECTION FIFO
45

FIG. 4


CELL DEPORTS
50
51
CONNECTION BACKLOGGED?
YES
DEQUEUE FROM CONNECTION
52
54
53
55
ENQUEUE ON SORTING BIN
NO
CONFORM?
YES
ENQUEUE ON TXFIFO

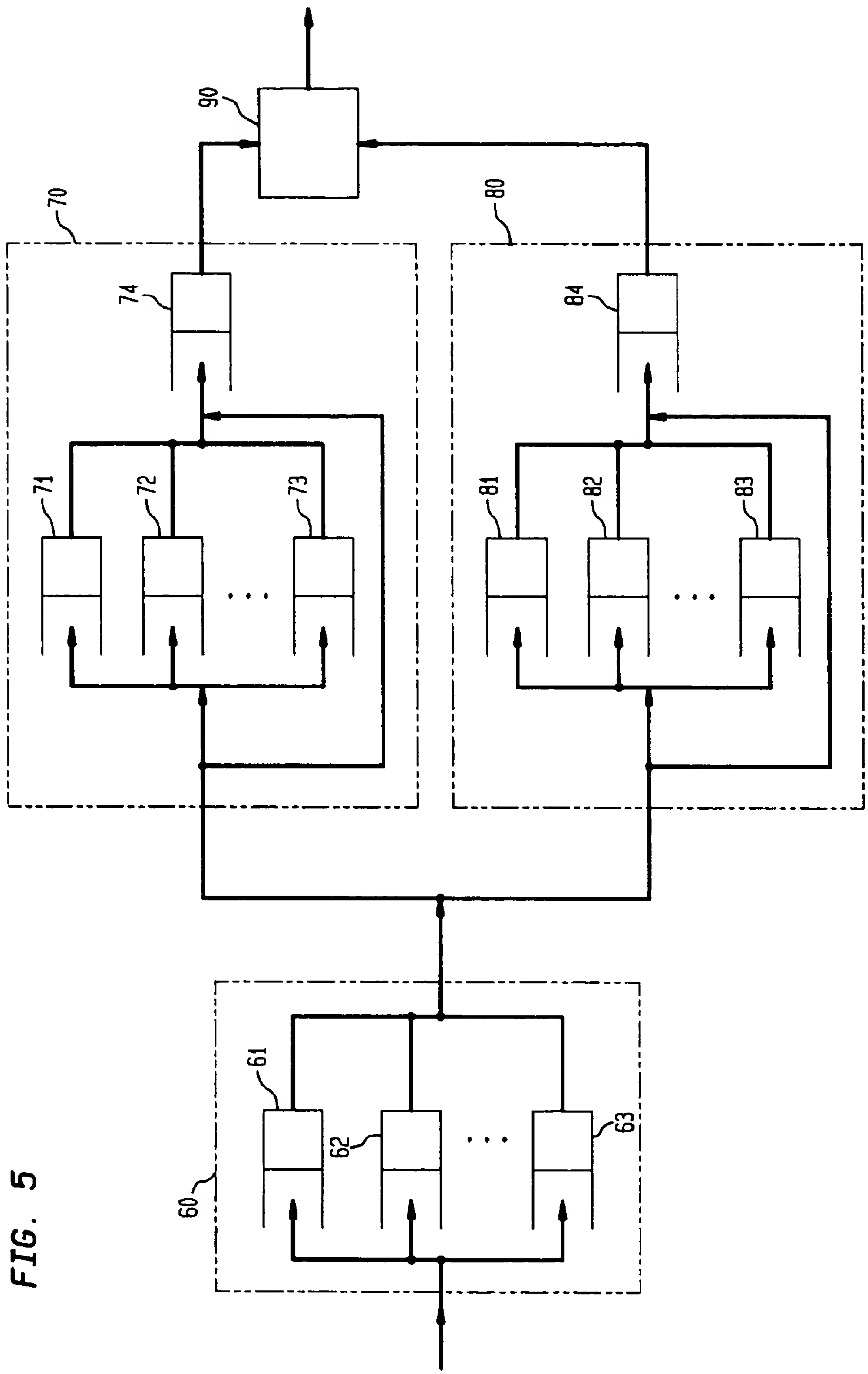
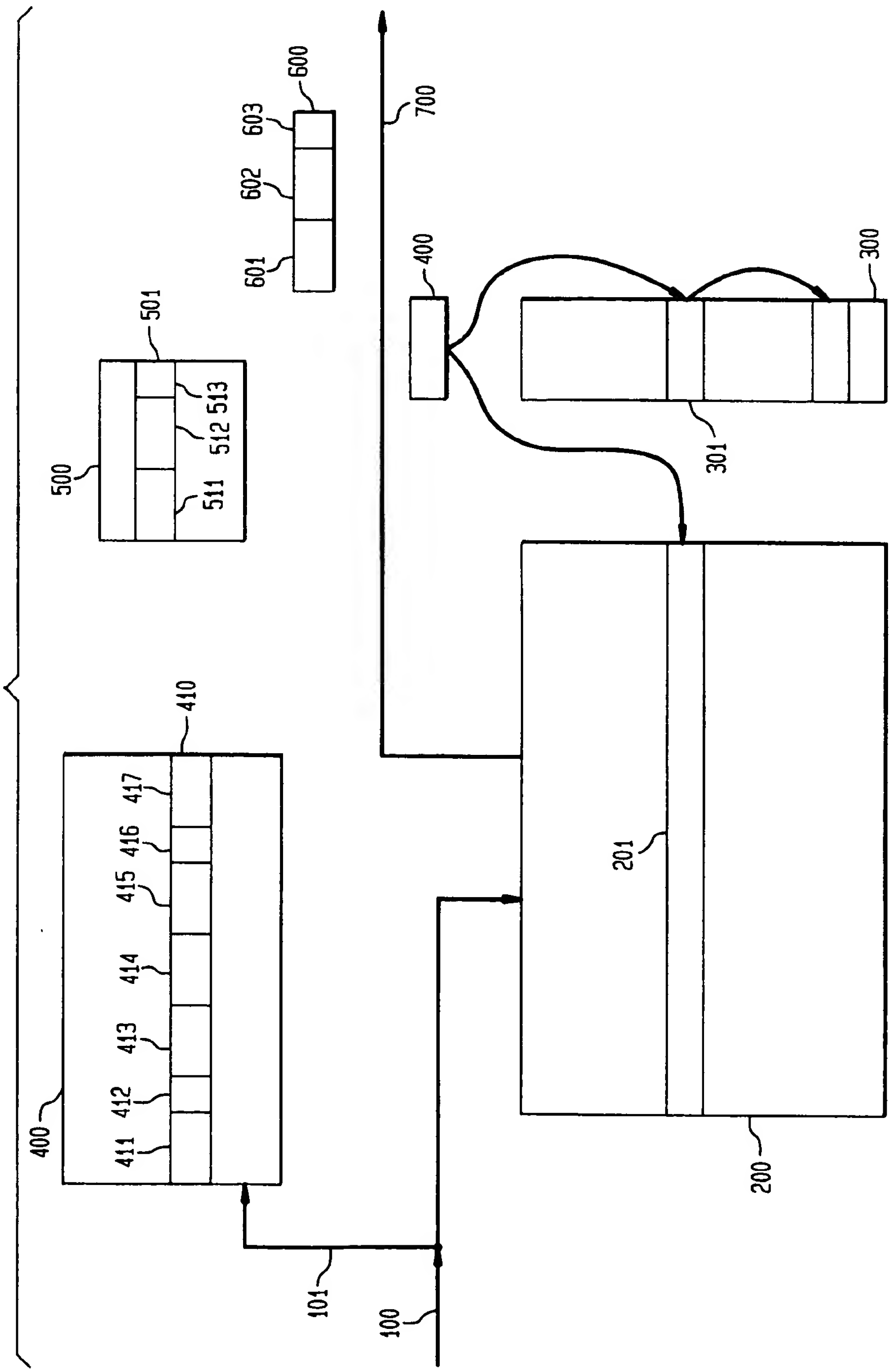


FIG. 5

FIG. 6


600
603
602
601
700
400
300
301
500
501
511
512
513
410
400
411
412
413
414
415
416
417
201
200
101
100

METHOD FOR INTEGRATED TRAFFIC SHAPING IN A PACKET-SWITCHED NETWORK

FIELD OF THE INVENTION

The invention relates generally to controlling traffic in a communications network. More particularly, the invention relates to integrated traffic shaping in an asynchronous transfer mode (ATM) switch operating in a high speed network.

BACKGROUND OF THE INVENTION

Communications networks currently transfer vast quantities of information in both local and wide area networks The information typically consists of signals (electronic or optical) representing digitized or digital voice, video, and/or data that is transferred between endpoints in networks. For information to be transmitted in a network, a communication path must be established within the network between the sender(s) and receiver(s) of the information. A communication path may be established by circuit switching, wherein an exclusive channel is established between the sender(s) and receiver(s) throughout the entire transmission and until the connection is released. Alternatively, a communication path may be established by packet switching, wherein virtual circuits or channels are established between sender(s) and receiver(s) and with the channel only occupied for the duration of transmission of the packet.

Packet switching enables networks to handle the heterogenous mix of network traffic with varying service requirements that are encountered in Broadband Integrated Services Digital Networks (B-ISDN). Ideally, packet switching is scalable and can reliably establish and maintain virtual channels without any prespecified rates (so-called bandwidth on demand). Asynchronous Transfer Mode (ATM) is a connection oriented network technology that provides a possible framework for ideal packet switching that is designed to support multiple classes (e.g., voice, video, data) of traffic.

In ATM, information is transmitted as packets of digital information called cells. Each cell includes 53 bytes of digital information, ATM is connection oriented; a connection is formed between the transmitter(s) and receiver(s) where each intermediate switch in the virtual circuit or channel (VC) is aware of the service requirements and traffic parameters of the connection. The links between switches are "virtual" only in that each link may be shared by several connections on a demand basis instead of a fixed allocation of the entire link to a single connection as in a circuit-switched network. By making each switch aware of the service requirements and traffic parameters for each connection, the quality of service (QOS) of the channel can be guaranteed by the network if the connection stays within its stated traffic parameters.

The quality of service in an ATM-switched network typically refers to the probability of a cell being dropped, (cell loss) and factors affecting cell delivery timeliness, particularly cell delay and cell delay variation. Traffic parameters typically refer to the rate at which data bits are transmitted through the network (bit rate) and the variations in the bit rate (burstiness). Different classes of traffic require different levels of QOS and have different traffic parameters. For example, voice communications are typically transmitted at a continuous bit rate (CBR) of 64 Kbit/second with no burstiness and can tolerate some cell loss but little delay. Another example is compressed packetized voice or video

which is transmitted at a variable bit rate (VBR) with varying degrees of burstiness and bounded limits on delay because of the need to reconstruct the video or voice. Computer file transfer and data network applications, on the other hand, will generate data at widely varying rates without stringent requirements regarding cell delay and may be readily transmitted whenever bandwidth is available in the channel at the available bit rate (ABR). Available Bit Rate traffic may also be classified as Unspecified Bit Rate (UBR) if no minimum cell rate is declared.

The QOS of a connection in an ATM switched network may be "guaranteed" by "contract" when a connection is established through a process of connection admission control (CAC). Essentially, each connection "contracts" to transmit cells to the network at a rate ρ (bandwidth descriptor) with burstiness σ (burst descriptor) when the connection is established. The network will not allow the connection to be established if there are insufficient network resources (e.g., buffer and/or bandwidth) to provide the required QOS at the contracted traffic parameters.

Once connections have been established, the connections have to be regulated to prevent congestion in the network. Although each connection is expected to comply with the contracted traffic parameters, it is necessary for the network to ensure compliance by the connection. Prior art techniques for regulating connections (traffic control) include policing the connection and discarding excess cells (i.e., cells violating the contracted traffic parameters) seeking to enter the network. See, e.g., Boyer, P., "A Congestion Control for the ATM," Proc. 7th ITC Spec. Seminar, Paper 4.3, New Jersey, October, 1990 (Describing pick-up policing algorithms). Prior art techniques have also suggested policing by tagging the cell loss priority (CLP) bit of non-compliant cells and only discarding the cells if the network becomes sufficiently congested to adversely affect network performance.

Traffic shaping is another form of traffic control for ATM switched networks. Traffic shaping can be used to ensure compliance with the traffic contract when applied at the edge of the network or, if applied within the network interior, to normalize traffic and limit jitter by reshaping connections and spacing cells. So-called "leaky bucket traffic shaping" is often used in ATM switched networks. In a typical prior art leaky bucket traffic shaper, credit "tokens" are provided by the traffic shaper at a rate ρ that represents the sustainable bandwidth for the connection. For example, tokens could be provided at a rate of one token every ten time slots, where a time slot represents the minimum time for a cell to leave the traffic shaper. In other words, a fully utilized traffic shaper would transmit one cell every time slot. These credit tokens accumulate in a "leaky bucket" that holds up to σ (burst size) credit tokens. Each cell arriving at the traffic shaper must claim a token to pass through the shaper; if no tokens are available, the cell is considered "nonconforming" and is delayed until a token is available when the connection conforms to the traffic contract.

A traffic shaper determines whether an arriving cell conforms to its connection's traffic parameters or descriptors using any of a variety of algorithms known in the art, the most common of which is the generic cell rate algorithm (GCRA) from the ATM Forum. The GCRA computes conformance based on a connection's shaping rate and burstiness and when the last cell arrived at the shaper from that connection.

Cells are scheduled for transmission from a traffic shaper based on the cell's conformance times. In typical prior art traffic shapers, cells are sorted in exact order of conformance

times. One prior art implementation is a shift register wherein the sorted cells (or pointers) are stored and appropriately shifted when a cell with an intermediate conformance time is added. This implementation is disadvantageous because of the prohibitive hardware costs incurred by parallel comparison operations and additional circuitry needed to compensate for clock rollovers.

An alternative prior art traffic shaper avoided these drawbacks by implementing the scheduler as a series of bins corresponding to single transmission slots. By traversing the bins at the transmission, the traffic shaper can transmit one cell at each time slot if there is a cell awaiting transmission in the bin. If the bin is empty then a cell will not be transmitted during that time slot. This traffic shaper puts an arriving cell into the first bin that corresponds to or follows its conformance time. There are, however, several disadvantages with this implementation. Complex logic is required to locate an appropriate empty bin and, under heavy traffic conditions a cell may be scheduled for transmission far after its conformance time. Moreover, early arrivals inherently receive preferential treatment at the expense of cells arriving from connections adhering to traffic contract parameters.

Some of these disadvantages have been addressed by another prior art traffic shaper using a calendar queue where each bin corresponds to a linked list of cells having a particular conformance time instead of a single cell. In this implementation, a cell is merely appended to the linked list corresponding to its conformance time. Because only one cell may be transmitted at a time, an additional transmit queue is required to enqueue multiple cells for output. This implementation does reduce the need for a complex exact priority queue, but also has several disadvantages.

The primary disadvantage is the need to accomodate the maximum possible number of conformance times while accounting for switch capacity limitations and the resultant possibility of dropping non-conforming cells. Although it has been suggested that a range of consecutive conformance times could be associated with each bin to reduce the number of bins, (i.e., bin granularity>1), large bin granularity creates excessive jitter distorting traffic parameters, especially in high bandwidth connections. Moreover, this implementation does not mediate between multiple connections with simultaneously conforming cells.

A traffic shaper with multiple incoming connections can quickly develop a backlog of conforming cells simultaneously eligible for transmission. Prior art techniques for arbitrating amongst the simultaneously conforming cells have proven inadequate, frequently increasing cell-shaping delays and distorting connection parameters for all cells, even cells that conformed upon arrival. These problems can propagate through the network, violating traffic descriptors for downstream switches and further increasing delay and loss. Constant bit rate and real-time variable bit rate connections are particularly impacted because of the inherent low tolerance for delay variations.

Prior art traffic policing and shaping techniques are proving to be inadequate for B-ISDN and the heterogenous traffic of contemporary networking, resulting in traffic distortions and congestion. In particular, the prior art techniques do not efficiently handle numerous traffic classes while fairly distributing network resources amongst several connections. The unique difficulties encountered when attempting to shape traffic from several connections with widely varying rates are not adequately handled by prior art policers and shapers. For example, prior art policers are unsuitable for responding to momentary violations of contract parameters by connections that are fully compliant over the long term, resulting in avoidable packet loss or traffic disruption. Many proposed techniques for fairly distributing network resources cannot be readily implemented because they are not readily scalable or impose unacceptable costs (e.g., switching delays or hardware implementation complexity) Without efficient and scalable techniques for traffic shaping, the potential of ATM switched networks cannot be realized.

SUMMARY OF THE INVENTION

In view of the foregoing, there is a need for scalable traffic shaping in a communications network that fairly distributes network resources between widely variant connections without incurring excessive costs.

The present invention simultaneously reduces implementation complexity and traffic distortions by a novel integration of leaky-bucket traffic shaping and rate-based link scheduling. The present invention is able to shape traffic at the edges of a network while equitably distributing bandwidth, even during periods of bursty traffic Moreover, the present invention can also schedule links in a network interior, reducing cell delay jitter and downstream buffering requirements.

Traffic shaping in the present invention is achieved through a novel combination of techniques. The initial step is to separately enqueue each connection coming into the switch. Per-connection queues are more than then mere extensions of sorting bins because they provide a mechanism for additional network management features such as buffer management and flow control. Moreover, per-connection queuing enables each connection to be treated separately, allowing the traffic shaper to take each connection's unique traffic parameters into account when shaping traffic received from that connection. By so doing, it is possible to perform weighted scheduling that provides preferential treatment to connections with critical parameters. Weighted scheduling ensures that high bandwidth connections or connections with no delay tolerance (e.g., real-time video) are able to maintain QOS when switched with low bandwidth or ABR connections.

Per-connection queuing alone will not optimally shape network traffic. Another aspect of the present invention is "approximate sorting," ensuring that cells leave the traffic shaper approximately, but not exactly, in order of conformance time. This is unlike prior art systems that perform exact sorting, with each cell leaving the traffic shaper having a conformance time equal to or greater than the conformance time of the preceding cell. Approximate sorting, like exact sorting, can be implemented with an output queue and sorting bins, but it differs from exact sorting in that the output queue can be a simple first in first out (FIFO) queue without requiring complex parallel comparison and insertion circuitry.

Approximate sorting may be implemented by moving a cell from a per-connection queue directly to an output queue if the cell's conformance time has been reached or passed and moving cells that are non-conforming into sorting bins until they conform, i.e., the cell's conformance time is reached. A cell that is in a sorting bin at its conformance time goes directly to the output queue at that time. Cells that are not moved from the per-connection queue into the output queue until after their conformance time will not be placed into the output queue at a location exactly corresponding to their relative conformance time, even though these cell bypass the sorting bins. Hence, the output queue will only be approximately sorted.

Fair arbitration amongst the incoming connections is another advantage of the present invention. There are several alternative techniques for determining when a cell from an incoming connection is processed by the traffic shaper in the present invention. The first technique is round robin arbitration, whereby a cell coming from a connection is processed every time a cell from that connection leaves the traffic shaper. During periods of heavy traffic, each incoming connection is essentially treated equally with each connection being serviced in sequence. The primary disadvantage with this technique is that backlogged high bandwidth connections process cells through the traffic shaper at the same rate as backlogged low bandwidth connections, making it difficult for the high bandwidth connections to maintain quality of service, even though overall shaping delays and traffic distortions are reduced.

The present invention addresses this problem by weighted round-robin scheduling. In weighted round-robin scheduling, some connections receive preferential treatment to ensure that more cells from the preferred connections are being processed by the traffic shaper at a given time. Weighted round-robin scheduling essentially operates by allowing several cells from a high bandwidth connection in the traffic shaper sorting unit simultaneously. Instead of processing one cell from the per-connection queue at a time, several cells are processed simultaneously. The scheduling is weighted in favor of the high bandwidth connections, ensuring that each connection has a quantity of cells in the output queue proportionate to the connection's bandwidth. This enables each connection to provide an appropriate quality of service during periods of high traffic, even if every connection is backlogged.

Weighted round-robin scheduling addresses one aspect of the difficulties created when low and high bandwidth incoming connections share a traffic shaper, but other difficulties remain. One of the difficulties inherent when connections with widely varying rates are common to a single traffic shaper is the wide variance in conformance time intervals The interval between conformance times for a low bandwidth connection is much larger than the interval between conformance times for a high bandwidth connection The possible conformance times for a low bandwidth connection may extend far into the future, especially relative to the possible conformance times of high bandwidth connections. It is frequently impractical to have sorting bins corresponding to the conformance times of both the high bandwidth connections (with short intervals between the sorting bin conformance times) and the low bandwidth connections (with lengthy intervals encompassing several high bandwidth conformance time intervals), and increasing bin granularity distorts traffic. However, the hierarchical architecture of the present invention provides a solution to these problems that avoids traffic distortions. Grouping connections based on their bandwidth requirements provides high bandwidth connections with small granularity sorting advantageously reducing jitter and cell shaping delay without requiring a prohibitively large number of sorting bins. Moreover, the weighted round-robin scheduling techniques of the present invention are equally applicable to arbitration between groups, further reducing traffic distortion and improving traffic shaper performance.

The advantages and alternatives of the integrated traffic shaper disclosed herein is described in J. Rexford et al., "A Scalable Architecture for Fair Leaky-Bucket Shaping," to be published in Proceedings, IEEE INFOCOM, April, 1997; and J. Rexford et al, "Scalable Architectures for Integrated Traffic Shaping and Link Scheduling in High-Speed ATM Switches," to be published in IEEE Journal on Selected Areas in Communications, Special Issue on Advances in ATM Switching Systems, 1997, the disclosures of which are incorporated by reference herein.

This invention provides an integrated system for shaping traffic in packet-switched networks that ensures resources are fairly shared between all switched connections while maintaining minimum QOS for each connection. This unique integration of novel techniques forms a traffic shaper capable of providing traffic shaping for the variety of network traffic classes possible in ATM networks. In contrast to prior art traffic shaping techniques, the present invention arbitrates fairly between all incoming connections and is highly scalable, thereby enabling Asynchronous Transfer Mode networks to be used to their full potential.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic representation of a scalable shaping architecture according to an embodiment of the present invention.

FIG. 2 is a flowchart representation of a procedure for determining the conformance time of an arriving cell.

FIG. 3 is a flowchart representation of a procedure for traffic shaping upon cell arrival.

FIG. 4 is a flowchart representation of a procedure for traffic shaping upon cell departure.

FIG. 5 is a schematic representation of a hierarchical scalable shaping architecture according to the present invention.

FIG. 6 is a block diagram representation of a hardware implementation of an embodiment of the present invention.

DETAILED DESCRIPTION

A novel traffic shaper for use in a packet switched network is described herein. This traffic shaper is optimally suited for, but not limited to, high speed switches in an Asynchronous Transfer Mode network. The disclosed traffic shaper is a novel implementation of per-connection queuing and approximate sorting techniques suitable for shaping traffic arriving from several connections with varying bandwidth and traffic characteristics such as burstiness. This novel implementation is highly scalable, requiring only a fixed number of operations to process a cell upon arrival or departure from the traffic shaper. By integrating link scheduling into the traffic shaper, the present invention is able to fairly allocate network resources between incoming connections in a novel manner that guarantees a minimum Quality Of Service for each connection. A novel hierarchical implementation of the present invention is disclosed by which traffic from connections with widely variant bandwidth requirements can be fairly shaped without excessive hardware requirements. An efficient hardware implementation of the present invention is also disclosed.

FIG. 1 illustrates the scalable architecture of the novel traffic shaper In one embodiment of the present invention, cells enter the traffic shaper through input **1** and are initially routed to a set of connection queues **10**. The set of connection queues **10** interface to a sorting unit **20** across connection **2** and the cell is ultimately transmitted to the network through output **3**. Although only three connection queues **11**, **12** and **13** are shown in FIG. **1**, there is a queue for each connection coming into the traffic shaper, e.g., a traffic shaper with ten incoming connections would have ten queues. In a preferred embodiment of the present invention the connection queues are first in first out linked lists (FIFOs).

Cells that arrive from an idle connection pass directly through the connection FIFOs **10** and across connection **2** to sorting unit **20**. If the connection is not idle, i.e., if a previous cell from the connection is in connection FIFOs **10** or sorting unit **20**, the newly arrived cell is enqueued on the appropriate connection FIFO. A cell entering sorting unit **20** will be enqueued on a sorting bin **21**, **22** or **23** if it is non-conforming or enqueued on an output queue **24** if it is conforming.

A cell is conforming if its conformance time c has been reached or passed (i.e., $c \leqq t$). In the preferred embodiment, cells are not placed in the transmission queue in exact order of conformance times. Instead, transmission queue **24** is a FIFO which is approximately sorted by conformance times. Cells entering sorting unit **20** from connection FIFOs **10** after their conformance time are not inserted into transmission queue **24** at the exact position corresponding to their conformance time but are instead appended to the end of transmission queue **24**. Although a cell entering sorting unit **20** at its conformance time is also appended to the end of transmission queue **24**, this operation enqueues the cell at the same location exact sorting would since no cells with later conformance times are in transmission queue **24**. Cells enqueued from a sorting bin are similarly positioned in transmission queue **24** at the exact sorting location (assuming bin granularity of 1).

The approximate sorting of the present invention advantageously reduces implementation complexity in the traffic shaper. For example, parallel insertion capability is not required in the transmission queue. More significantly and perhaps unexpectedly, approximate sorting performs better than exact sorting for connections with similar bandwidth characteristics by guaranteeing a minimum service rate for each connection on a small time scale.

The number of sorting bins required for the sorting unit is advantageously independent of the number of incoming connections or the amount of backlogged traffic at the switch. The granularity of the bins (i.e, the number of consecutive conformance times associated with a single bin) and the shaping rate of the slowest incoming connection (i.e., the minimum shaping rate, ρ_{min}) determine the maximum number of sorting bins required for the illustrated embodiment of the present invention. The optimal number of sorting bins for a sorting unit in the present invention varies depending on the divergence of incoming connection bandwidth parameters; the lowest shaping rate determines the range and the highest shaping rate determines the grain.

The bounded number of sorting bins is a product of per-connection queuing. A cell arriving in sorting unit **20** will have a conformance time $1/\rho$ or less time slots in the future. Cells with conformance times farther in the future will be enqueued in a connection FIFO until the previous cell from that connection has been transmitted. For example, if a cell in sorting unit **20** was transmitted at its conformance time, the next cell from the same connection will necessarily conform in $1/\rho$ time slots (or less if the connection has unused σ) and would be placed in the corresponding sorting bin.

Because of the bounded number of sorting bins and the reduced complexity of a single approximately sorted transmission FIFO, the present invention has a highly scalable architecture. The scalability of the present invention is further enhanced because a small, bounded number of operations are performed each time a cell arrives at or departs from the novel traffic shaper. Thus, only the number of connection queues increases in direct proportion to the number of connections to the traffic shaper.

The first thing that occurs when a cell arrives at the traffic shaper is computation of the cell's conformance time c. Conformance time is the time at which an arriving cell conforms to the contracted traffic parameters of the connection the cell is coming from. These traffic parameters represent a negotiated agreement between the network and the connection whereby the network agrees to provide a specific quality of service and the connection agrees to only transmit a specific number of cells in a given interval. The rate at which the connection agrees to transmit cells to the network is the bandwidth or shaping rate ρ. In addition, the network may agree to allow the connection to transmit bursts of cells exceeding the negotiated shaping rate. The number of negotiated excess cells is the burstiness, σ, of the connection. In one embodiment of the present invention, conformance time is computed using the generic cell rate algorithm of the ATM Forum as illustrated by the flowchart of FIG. **2**.

A cell's conformance time is determined from the cell's estimated arrival time, X, the contracted traffic parameters for the cell's connection, ρ and σ, and the arrival time of the last cell from the same connection. Referring to step **31** of FIG. **2**, if a cell has not been received from a connection for a relatively long period of time or the traffic shaper has just been initialized, the state variable X, representing the cell's arrival time, is initialized to $-1/\rho$ to avoid clock wrap-around errors. If X is not reinitialized then it corresponds to the arrival time of the previous token received from the connection. The cell's arrival time is estimated at step **32**. A cell's estimated arrival time is $X+1/\rho$ or the current time, t, whichever is greater. Whether the estimated arrival time X complies with the traffic contract is determined at step **33** where X is compared to $t+1/\rho$. If X is less than or equal to $t+\sigma/\rho$ then the connection is complying with the traffic contract and the cell is conforming as shown in step **34**. In the case of a conforming cell, conformance time c equals the current time t. As shown in step **35**, if X is greater than $t+\sigma/\rho$ the cell is non-conforming and the conformance time is set to comply with the contracted traffic parameters, $c = X-\sigma/\rho$.

The operations performed when a cell arrives at the traffic shaper are shown in FIG. **3**. As noted above, the first operation performed is computation of cell conformance time, c, as shown in step **30**. As shown in step **41**, the next operation is a determination of whether the connection is currently idle, i.e., there are no cells from the connection enqueued in the connection FIFO or the sorting unit. As shown in step **42**, if the cell is idle, then a determination is made as to whether the cell is conforming. If the cell conforms, $c == t$, the cell is enqueued on the transmission FIFO as shown in step **43**. If the cell is non-conforming, the cell is enqueued on a sorting bin as shown in step **44**. In an embodiment having b sorting bins of grain g in the sorting unit, the cell is enqueued onto sorting bin [(c mod bg)/g]. If the connection is not idle, the cell is enqueued on the connection FIFO as shown in step **45**.

The operations performed when a cell departs from the traffic shaper are shown in FIG. **4**. When a cell received from a connection departs as shown in step **50** a determination is made as to whether the departed cell's connection is still backlogged as shown in step **51**. If the connection is not backlogged then the connection is idle and no further operations are performed. If the connection is still backlogged then a cell is dequeued from the connection FIFO as shown in step **52**. A determination is made as to whether the cell is conforming, $c \leqq t$, as shown in step **53**. If the cell is non-conforming, it is enqueued on a sorting bin as shown in step **54**. In the embodiment having b sorting bins of grain g in the sorting unit, the cell is enqueued onto sorting bin [(c

mod bg)/g]. If the cell is conforming then it is enqueued on the transmission FIFO as shown in step **55**.

An additional operation performed during each time slot is appending the conforming cells enqueued in a sorting bin onto the transmission FIFO. In the embodiment having b sorting bins of grain g in the sorting unit, sorting bin [(t mod bg)/g] is appended to the transmission FIFO when the cells in the sorting bin are conforming, i.e., when (t mod g)==(g−1). This is a background operation, the concatenation of two linked lists, and advantageously does not initiate any data transfers in the present invention.

An alternative embodiment of the present invention which does not require time stamps for each arriving cell is possible because of approximate sorting. Unlike exact sorting leaky-bucket traffic shaping, in which the conformance time of every head-of-line cell must be determined so a cell can be placed into its exactly sorted position in the output queue, approximate sorting only requires calculation of non-conforming head-of-line cell conformance times. However, non-conforming head-of-line cell conformance times can be readily determined by the traffic shaper if it maintains per-connection state information, specifically the value of the state variable X when the first non-conforming cell arrives. This saved value, X_b, is the state of the "boundary cell" at the boundary between conforming and non-conforming cells. Subsequent non-conforming cells will be spaced 1/ρ apart and their conformance times can therefore be readily computed from X_b. When a conforming cell subsequently arrives from the connection all previously received cells will necessarily be conforming and X_b can be reset to null. This ensures that there is only one boundary cell per connection at any time, even if the shaper is unable to keep up with conforming traffic. Thus, by determining the boundary between conforming and non-conforming cells, this alternative embodiment is able to eliminate calculation and storage of conformance time for each buffered cell.

This alternative embodiment can also advantageously respond to dynamic variations in connection bandwidth parameters. First, the traffic shaper can implicitly adjust cell conformance times by applying a new value of ρ when computing conformance time c and updating the boundary state X_b. Since the current state variable X is calculated from X_b and the number of non-conforming cells enqueued in the connection FIFO, the traffic shaper can readily adjust the current state variable X to account for changes in the shaping rate ρ. The state variable X is adjusted as follows: $X \leftarrow X_b + (X - X_b)\rho^{old}/\rho^{new}$. The new rate ρ^{new} is therefore advantageously applied to nonconforming cells that haven't entered the sorting unit instead of waiting until the connection backlog has been processed by the traffic shaper.

The cell departure operations described above with respect to FIG. **4** ensure that incoming connections are serviced on a round-robin basis when the incoming connections are backlogged. Transmitting cells from the transmission FIFO causes the output link to implicitly sequence through these connections. This procedure most fairly arbitrates between incoming connections when the incoming connections have similar bandwidths. Another embodiment of the invention addresses where incoming connections have widely varying bandwidths.

The alternative embodiment uses weighted round-robin scheduling whereby each connection receives service in proportion to its bandwidth descriptor ρ. Instead of only allowing one cell from a connection in the sorting unit (sorting bins and transmission FIFO) at a time, each connection could have a number of cells in the sorting unit. The number of cells allowed would be proportionate to the connection bandwidth and the bandwidth of the slowest incoming connection. For example, connection α would have 0 to ρ_α/ρ_{min} cells in the sorting unit at a times where ρ_α is the shaping rate of connection α and ρ_{min} is the shaping rate of the incoming connection with the lowest bandwidth. By integrating novel rate-based fair link scheduling and traffic shaping in this manner, the present invention advantageously ensures that each incoming connection receives sufficient bandwidth on a small time scale. The novel integrated traffic shaper is thus able to maintain quality of service for connection traffic and reduce traffic distortions without increasing the number of sorting bins or inhibiting the scalability of the traffic shaper.

FIG. **5** illustrates another alternative embodiment of the present invention whereby several incoming connections with widely varying bandwidth parameters without adversely affecting high bandwidth traffic. As discussed above, low bandwidth connections will input cells into the traffic shaper that have conformance times far ahead of cells from high bandwidth connections. Thus, low bandwidth connections require sorting bins capable of handling a wide range of conformance times. Although it is possible to reduce the number of required sorting bins by using sorting bins with a high granularity g, highly granular or coarse sorting bins are unsuitable for high bandwidth connections because of the shaping delay and jitter they introduce in high bandwidth traffic.

The alternative embodiment of the present invention resolves this dilemma by a novel hierarchical traffic shaper architecture in which connnections are grouped by bandwidth requirements. As shown in FIG. **5**, the traffic shaper includes a set of connection FIFOs **60** but, instead of a single sorting unit, there are separate sorting units **70** and **80**. As in the non-hierarchical architecture of FIG. **1**, the set of connection FIFOs **60** is simply a set of queues **61**, **62** and **63** with one queue for each incoming connection. Similarly, each sorting unit **70** and **80** consists of sorting bins (**71**, **72** and **73** or **81**, **82** and **83**) and a transmission queue **74** or **84**. Although only two sorting units are explicitly shown in FIG. **5**, the present invention is not so limited.

Each separate sorting unit with bins of grain g and range bg corresponds to a separate group of connections having similar bandwidth parameters. The separate groups of the hierarchical architecture advantageously enables optimization of the sorting unit parameters g and bg for a specific range of bandwidths without requiring an excessive number of sorting bins or distorting traffic.

As shown in FIG. **5**, the hierarchical architecture of the alternative embodiment requires a group arbiter **90** to multiplex the cells from the separate sorting unit transmission queues and fairly arbitrate between the groups. Fair arbitration between the groups may be efficiently implemented on the basis of group weight ϕ using rate-based link scheduling algorithms known to those of ordinary skill in the art. Essentially, each group i has a weight ϕ_i representing the aggregate bandwidth requirements of the group. For example, ϕ_i may equal the sum of ρ_α across all connections α_i in group i. A fair portion of the link bandwidth would be $\phi_i/\Sigma\phi_j$, for j=0 to n−1, where n is the number of groups.

In the embodiment of the present invention where the group weight is static, i.e., $\phi_i = \Sigma\rho_{\alpha i}$, the unused bandwidth from an idle connection is divided between the busy connections in the same group and is not available to other connections in other groups. This is particularly advantageous when each group corresponds to a specific traffic class

or protocol family because each connection has a guaranteed bandwidth through the traffic shaper.

In another alternative embodiment of the present invention, group weights are dynamic. In this embodiment, $\phi_i(t)=\Sigma\rho_{\alpha i}$ for $\alpha_i \epsilon B_i(t)$ where B_i is the set of backlogged connections in group i at time t, and bandwidth is shared between individual connections by altering the group weight ϕ_i as a function of aggregate throughput requirements of backlogged connections. The set of backlogged connections, $B_i(t)$, and thus the group weight, ϕ_i, changes dynamically as cells arrive and depart and connections are idled or backlogged. For example, if a connection α_i is backlogged then $\phi_i=\phi_i+\rho_{\alpha i}$ whereas if a connection α_i is idled then $\phi_i=\phi_i-\rho_{\alpha i}$.

The dynamic group weights may be used advantageously to divide unused bandwidth between all incoming connections in a hierarchical architecture by extending known fair scheduling algorithms. For example, self-clocked fair queuing algorithms known to those of ordinary skill in the art can be readily modified for dynamic group weights. A self-clocked fair queuing algorithm arbitrates on the basis of virtual finishing times, F_i, that is computed using a simple recurrence and closely approximates weighted fair queuing algorithms. In essence, the traffic shaper arbitrates by transmitting the cell from the backlogged group with the minimum F_i in each transmission slot. The virtual finishing time F_i reflects bandwidth consumed by the group i by incrementing F_i by $1/\phi_i$ after each transmission from group i. Fair arbitration occurs by assigning a value of $F_i=F^{serv}+1/\phi_i$, where F^{serv} is the virtual finishing time of the group currently in service, whenever group i is backlogged. Previously idle groups are thus precluded from taking credit for past bandwidth.

An efficient hardware implementation of the present invention is illustrated in FIG. 6. In this implementation, the connection queues, sorting bins and transmission queues are implemented as linked lists of cells buffered in a cell memory with associated pointer fields.

Referring to FIG. 6, a cell enters the traffic shaper at input **100**, a serialized stream of cells from one or more incoming connections. The arriving cell is buffered in cell memory **200** at memory location **201**.

Each buffered cell includes a pointer field **301** in an array of pointers **300** that denote each location in the cell memory. A free list **400** is also maintained in the shaper for assigning available memory locations to arriving cells. In a preferred embodiment, the free list is maintained as a last-in first-out list using the pointers **300** as shown in FIG. 6. For example, if cell memory location **201** was used by the last cell to leave the traffic shaper then the arriving cell would be buffered in memory location **201** denoted by pointer **301** associated with free list **400**.

A connection identifier **101** is associated with each arriving cell and processed by the corresponding connection entry **410** in connection table **400**. Each connection entry **410** includes fields that describe the connection and maintain the associated connection FIFO.

The connection descriptor fields are burst tolerance, σ/ρ, **411**, shaping rate, ρ, **412**, and state variable, X, **413**. As discussed above, the values in these fields are used to determine conformance time of cells arriving from the connection corresponding to connection entry **410**.

Each connection queue is implemented as a linked list FIFO in the embodiment of FIG. 6. Thus, the connection FIFO fields include a head pointer **414** and a tail pointer **415**. The head pointer **414** points to the location of the buffered cell that is at the head of the connection queue while the tail pointer **415** points to the location of the buffered cell at the end of the connection queue. An empty flag **416** indicates whether there are any buffered cells buffered in the connection queue. In an implementation using round-robin arbitration, a flag **417** indicates whether any head-of-line cells from the connection are in the sorting unit (sorting bins or output queue) In an alternative implementation using weighted round-robin scheduling, register **417** is a counting semaphore indicating the number of cells from the connection that are in the sorting unit.

A cell buffered in cell memory **200** will only be enqueued in a connection FIFO if the connection is backlogged when the cell arrives. If the connection is not busy but the cell is non-conforming at arrival, the cell is enqueued onto a sorting bin **501** in sorting bins **500**. Each sorting bin is implemented as a linked list in the embodiment of FIG. 6. Thus, the sorting bin fields include a head pointer **511** and a tail pointer **512**. The head pointer **511** points to the location of the buffered cell that is at the head of the sorting bin linked list while the tail pointer **512** points to the location of the buffered cell at the end of the sorting bin linked list. An empty flag **513** indicates whether there are any buffered cells in the sorting bin.

If a cell is conforming upon arrival from an idle connection, the cell is enqueued onto the output queue **600**. The output queue is implemented as a linked list FIFO (transmission FIFO) in the embodiment of FIG. 6. The transmission FIFO fields include a head pointer **601** and a tail pointer **602**. The head pointer **601** points to the buffered cell at the head of the transmission FIFO and the tail pointer **602** points to the buffered cell at the end of the transmission FIFO. An empty flag **603** indicates whether there are any buffered cells in the transmission FIFO. In the embodiment shown in FIG. 6, the cell pointed to by the head pointer **601** is read from cell memory **200** and transmitted through output **700** at each time slot.

The integrated traffic shaper architecture shown in FIG. 6 further includes an arithmetic logic unit to perform the comparison and pointer manipulation operations of the present invention. In addition, a separate memory may be necessary to store cells awaiting service by the traffic shaper.

Because the data structures of the traffic shaper are implemented as linked lists, cells can be efficiently moved between logical FIFO queues by manipulating pointers. Moreover, using separate memory modules for each data structure as shown in FIG. 6 enables overlapping operations to be performed on different units, advantageously achieving greater concurrency in processing cell arrivals and departures.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, integrated scheduling may be based on delay or QOS parameters in addition to or in lieu of connection shaping rate. Alternatively, cell conformance could be determined on the basis of connection shaping rate and output scheduling determined on the basis of allowable connection delay. Moreover, the novel traffic shaping disclosed herein could be applied to a traffic shaper with several outputs and a common cell memory.

What is claimed is:

1. A method of shaping traffic in a packet switched network having at least one incoming connection and at least one outgoing connection, comprising the steps of:

enqueing packets received from an incoming connection on a queue associated with said incoming connection;

determining the conformance time of each packet received from an incoming connection;

transferring a packet at the head of said queue associated with said incoming connection to an output queue or to one of a plurality of sorting bins if said packet's conformance time is later than the current time;

transferring a packet from one of said plurality of sorting bins to said output queue when said current time precedes or is simultaneous with said packet's conformance time; and

transmitting a packet from said output queue.

2. The method of shaping traffic in claim **1** wherein each said queue associated with said incoming connection is uniquely associated with a single incoming connection.

3. The method of shaping traffic in claim **1** wherein each said queue associated with said incoming connection is a first in first out queue.

4. The method of shaping traffic in claim **1** wherein a shaping rate and a burst tolerance is associated with each said incoming connection and said step of determining the conformance time of a packet received from an incoming connection comprises the steps of:

estimating said packet's arrival time;

evaluating the chronological relationship between said packet's estimated arrival time and said packet's actual arrival time; and

determining said packet's conformance time based on said evaluation of said packet's estimated arrival time relative to said packet's actual arrival time.

5. The method of shaping traffic in claim **4** wherein said step of determining said packet's conformance time based on said evaluation of said packet's estimated arrival time relative to said packet's actual arrival time comprises the steps of:

determining said packet's conformance time is equal to said actual arrival time when said estimated arrival time precedes or is simultaneous with said actual arrival time; and

determining said packet's conformance time precedes said estimated arrival time by an interval corresponding to said burst tolerance and said shaping rate when said actual arrival time precedes said estimated arrival time.

6. The method of shaping traffic in claim **4** wherein a shaping rate and a burst tolerance is associated with each said incoming connection and said step of estimating said packet's arrival comprises estimating said arrival time as a function of the shaping rate and burst tolerance of said incoming connection and the arrival time of the last packet received from said incoming connection.

7. The method of shaping traffic in claim **1** wherein said step of transferring a packet at the head of said queue associated with said incoming connection to an output queue or to one of said plurality of sorting bins occurs when a packet associated with said incoming connection is transmitted from said output queue.

8. The method of shaping traffic in claim **1** further comprising the step of transferring a plurality of packets from said queue associated with said incoming connection to at least one of said plurality of sorting bins or said output queue.

9. The method of shaping traffic in claim **8** wherein a shaping rate is associated with each said incoming connection and the maximum number of packets associated with one of said incoming connections in said plurality of said sorting bins and said output queue is a function of the shaping rate of the incoming connection having the slowest shaping rate and the shaping rate of said one of incoming connections associated with said packets.

10. The method of shaping traffic in claim **1** wherein each of said sorting bins is a linked list.

11. The method of shaping traffic in claim **1** wherein said output queue is a first in first out queue.

12. The method of shaping traffic in claim **1** wherein each of said packets has a fixed size.

13. The method of shaping traffic in claim **1** wherein each said sorting bin corresponds to a single delay period between said conformance time and said current time.

14. The method of shaping traffic in claim **1** wherein each said sorting bin corresponds to a range of delay periods between said conformance time and said current times.

15. A method of shaping traffic in a packet switched network having a plurality of incoming connections and at least one outgoing connection, comprising the steps of:

enqueuing packets received from an incoming connection on a queue associated with said incoming connection;

determining the conformance time of each packet received from an incoming connection;

transferring a packet at the head of said queue associated with said incoming connection to one of a plurality of sorting units, each of said sorting units comprising at least one sorting bin and an output queue, wherein said packet is transferred to one of said sorting bins if said packet's conformance time is later than the current time;

transferring a packet from one of said sorting bins to one of said output queues when said current time precedes or is simultaneous with said packet's conformance time;

transferring a packet from one of said output queues to a group arbiter; and

transmitting a packet from said group arbiter.

16. The method of shaping traffic in claim **15** wherein said step of transferring a packet from one of said output queues to said group arbiter occurs at a frequency determined by a fair, rate-based scheduling algorithm.

17. The method of shaping traffic in claim **15** wherein said step of transferring a packet at the head of said queue associated with said incoming connection to one of a plurality of sorting units occurs at a frequency determined by a round-robin arbitration.

18. The method of shaping traffic in claim **15** wherein said step of transferring a packet at the head of said queue associated with said incoming connection to one of a plurality of sorting units occurs at a frequency determined by a weighted round-robin arbitration.

* * * * *